Volume 4, Issue 8 **September, 1982**

Price $1.50

TRS-80® Microcomputer News

Information Published for TRS-80 Users.

Education Issue

Color Planetarium

Fort Worth Scene



TANDY NEWSLETTER

We are currently using the Tandy Newsletter on CompuServe to make program listings from TRS-80 Microcomputer News available to you without your having to enter them. We don't have the staff to create a tape or disk version of the News, but we can provide the listings on CompuServe.

What you will find in the Tandy Newsletter (GO TRS-1 in CompuServe) are compact program listings (no extra spaces, etc.) which can be downloaded and run in the appropriate machine, assuming you have appropriate hardware and software. You will not find instructions, explanations or supporting material for the programs. This information is available in the appropriate issue of the News. We know that this is not a solution that every TRS-80 user will be able to take advantage of, but we want to make it available for those who can use it.

Disk Model I and III CompuServe users can download the material using the new Disk-based VIDTEX which is available from CompuServe. Model II users can download using the TRSDOS TERMINAL utility. Radio Shack does not currently offer downloading capabilities for either the Color or Pocket Computers.

We will be including all program listings which are more than ten lines in length in this service. We know that some of the material (such as for PC-1) can not be downloaded into the machine the program was designed to run in. We are making all of the programs available, in the hope that the people who can download the material will be more willing to take a program designed for a TRS-80 other than their own and modify it for their machine. If you do modify one of these programs, send us a copy of the changes (through CompuServe or on electronic media) and we may be able to run them in a future issue of the News.

We have also begun a feature on the CompuServe Tandy Newsletter which allows us to get corrections to published issues of the News to you very quickly. While we make every effort to produce error free issues of TRS-80 Microcomputer News, there always seem to be a few errors that find their way into print. The News correction section will give you a quick way of finding corrections. If you happen to locate an error which is not listed on CompuServe, please let us know so we can pass the information along to others.

A NEW WRITER

With this issue we are happy to welcome a new writer to the News staff. Annette Zamberlin-Main has joined us and we have probably already overloaded her with work. Annette will be working with both the News and with the Tandy Newsletter on CompuServe. Annette is largely responsible for keeping the Tandy Product Information section on CompuServe up to date. With her help we are expanding that section to include all Radio Shack products, not just Computer items.

New Computer Clubs

THE BASICS
Frederick Town High School
c/o Gary Mayton
117 Columbus Rd.
Fredericktown, OH 43019

JOHNSON AMATEUR COMPUTER SOCIETY
34 South Fir Street
Medford, OR 97501
Computer Bulletin Board System phone number
503-535-6883

SILICON VALLEY COMPUTER CLUB
GTE Systems Bldg. 3
Central Expressway and Whisman Rd.
Mountain View, CA

SOUTH BAY TRS-80 USERS GROUP (SBUG)
DYSAN Bldg.
5401 Patrick Henry Dr.
Santa Clara, CA

TRS-80 USERS GROUP
Harriette Gross
c/o John C. Hart Memorial Library
1130 Main St.
Shrub Oak, NY 10588

See Page 38 for More Computer Clubs.



Sam Rosen Elementary School
Fort Worth Independent School District

TRS-80 Microcomputer News: ©1982 Tandy Corporation, Fort Worth, Texas 76102 U.S.A. All Rights Reserved

Reproduction or use, without express written permission from Tandy Corporation of any portion of the Microcomputer News is prohibited. Permission is specifically granted to individuals to use or reproduce material for their personal, non-commercial use. Reprint permission for all material (other than William Barden's article), with notice of source, is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars ($12.00) per year, check or money order. Single copies of the of Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for $1.50 suggested retail each. The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars ($15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price $4.95 for the set). Back issues of 1981 copies are not available.

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted. If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you. If the material is very short, send it to us in E-Mail. If you have more than a few lines, you need to place the material in the ACCESS area of CompuServe and then let us know it is there by leaving a message on E-Mail.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007,535.

Notes to Program Users:

Programs published in the Microcomputer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack can not assume any liability for the accuracy either of the programs themselves, or of the results provided by the programs.

Further, while Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such can not be supported by our Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News. We will try (given the limited size of our staff) to find an answer to your question and, in many cases, will publish the answer in an up-coming issue of Microcomputer News.

Trademark Credits

CompuServe™	CompuServe, Inc.
Dow Jones™ NEWS/RETRIEVAL Service®	Dow Jones & Co., Inc.
ReformaTTer™	Microtech Exports
Time Manager™	Image Producers
VisiCalc®	VisiCorp, Inc.
AlphaKey™	Tandy Corporation
MicroPILOT™	Tandy Corporation
Program Pak™	Tandy Corporation
SCRIPSIT™	Tandy Corporation
TRSDOS™	Tandy Corporation
TRS-80®	Tandy Corporation

TRS-80® Microcomputer News

Volume 4, Issue 8 **September, 1982**

Price $1.50

Information Published for TRS-80 Users.

Contents:

Color Computer
- Program
 - Planetarium by J.W. Hurley 45

Computer Clubs 38

Computer Customer Service
- Computers in Education 8

Data Bases
- **CompuServe** 13
 - The Multiple Choice Puts You to the Test
- **Dow Jones** 14
 - Window on Wall Street
- **Profile** 43
 - Transferring Data From Records Into Shorter Records
- **VisiCalc** 17
 - Personal Time Savers

Education
- Figure It Out With Alice and Mac 12
- micro Research Problems 15
- News From the Education Division 30
- Straight Talk About Radio Shack Network Systems 29

Feature Story
- Radio Shack Network in the Classroom 5

Fort Worth Scene 2

General Interest
- Agriculture Source Book 44
- IRA Account by E. C. Hoover 7
- New Journal Calls For Papers 44
- Search and Rescue and the TRS-80 by J. Wellman 7

Model I/III
- Bugs, Errors, and Fixes
 - Accounts Payable (26-1554) 25
 - Accounts Receivable (26-1555) 26
 - Cassette Portfolio (26-1506) 25
 - COBOL Compiler (26-2203) 29
 - Disk Payroll (26-1556) 27
 - Medical Office System (26-1568) 27
 - Real Estate Vol. III (26-1573) 28
 - RUNCOBOL 28
 - Stockpak (26-1507) 25
- Programs
 - Arithmetic Practice by P. Fried 24
 - Crush - A Mathematical Game by T. Myerson 22
 - Customary to Metric by T. Pruzan 19
 - Directory Algorithm by T. Kelanic 20
 - Physical Science Program by A. Freeman 19
 - Prime Generator #1 by G. Greer 23
 - Prime Generator #2 by R. Euliano 23
 - String Formatting and the Window Game by D. Fuger 21

Model II
- Bugs, Errors, and Fixes
 - Accounts Payable (26-4505) 37
 - Accounts Receivable (26-4554) 38
 - BISYNC (26-4715) 38
 - General Ledger (26-4501) 36
 - Mailing List II(26-4507) 37
 - Payroll (26-4503) 37
 - RSCOBOL Generator (26-4707) 38
 - RUNCOBOL 36
 - Statistical Analysis (26-4540) 38
- Programs
 - Auto Field Routine by M. Lupul 33
 - Calculator by A. Banick, Jr 34
 - General Electric ASCII Sequential Files by R. Halloran 35
 - Read Screen by A. Pacheco 34
 - String Machine Language Programs by C. Hamkins 33

Peripherals 16
- Modem II Questions and Answers

Pocket Computer
- Start-Up Instructions 39
- Programs
 - Homework by Brian Cutler 39
 - Use of the TRS-80 Pocket Computer in Celestial Navigation by D. Parson 40

View from the 7th Floor by Jon Shirley 4

Prices shown in TRS-80 Microcomputer News are in U.S. Funds.

View From the Seventh Floor

by Jon Shirley
Vice President
Radio Shack Computer Merchandising

This month my column is devoted to the Color Computer. Those of you who do not own one may still enjoy reading this as there is a little fun for everyone here. I get several letters a week from Color Computer owners who complain that we do not understand that it is a very powerful machine, that we only make and advertise game software, that we do not give out any information on the product (where have I heard that before?), and that in general we totally neglect it. Rather than blunt my pen continuing to answer all these letters personally I would like to use my forum here to respond.

The first thing that I learned from these letters back in April is that the Color Computer is not a TRS-80 or a Color Computer but a CoCo. Up until then I thought a CoCo was the Cockatiel we had at home that escaped one Sunday to enjoy life in the wilds of Texas. All in all I must say that CoCo is a more flattering nickname than that other one we got a few years ago.

Look guys, we know the 6809 is the best 8 bit processor and that the CoCo is a very powerful computer. After all we did design the thing. In fact we worked very closely with our semi-conductor supplier to specify the SAM chip inside which connects the 6809 to the VDG and eliminates a whole mess of chips that would have cost more and made the board larger. The use of the 6809 and the SAM is what lets our graphics work so fast. In fact there are many bench mark programs that run as fast on a CoCo as on a 16 bit IBM P.C.

On the other hand we also designed the CoCo to be cheap, which is why I assume a lot of you bought it. Any computer involves a lot of design trade off's. If you design for low price you trade off features, yet we managed to include quite a few like on-board extended BASIC, RS-232C, etc. The primary market we saw, and still see, for the CoCo is in the home. And in addition to being cheap that means game software in ROM, easy to use, and as child proof as possible. And the more we sell because of this huge market, the more software and hardware we can bring out for the CoCo and the more we can lower the price as we did in July.

The major design trade off we made was in the use of the VDG (for those of you who are lost, the VDG is the Video Display Generator, an integrated circuit that generates everything you see on the screen) and it's 32 character by 16 line display and associated color set and graphics resolution. There are other ways to generate the screen, label them expensive. No upper and lower case is one of the missing items but there are some benefits. Unlike some of our competitors this is not an "exclusive" chip and all the information needed to do machine language programming is available. And frankly I think 32 by 16 beats 23 by 22 (or is it 22 by 23?)

The "you only advertise games" is one that surprises me. We have not only advertised the extended BASIC version a lot, but we also advertised our disk drives and ran an ad in BYTE and others with quotes from a variety of magazines and experts who all were saying what a powerful product the CoCo really is. And as far as software in general is concerned, it is NOT all on ROM paks (example the three cassette adventure games), and it is NOT all games (examples Scripsit, Spectaculator, Learning Lab, etc.). I will be the first to admit the disk software is late, but it is in the works and Spectaculator and Scripsit (with software upper and lower case) will be out soon to be followed by many more.

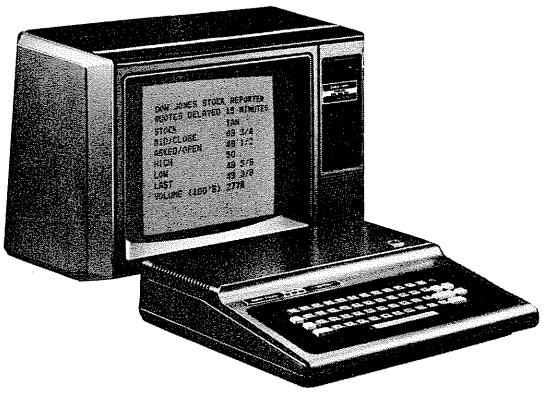
The annual catalog will feature two new peripherals designed for the CoCo and there are two more in the works. Also in the works are not one but two books by Mr. Bill Barden, the super author of Machine Language Programming on the TRS-80, who will tell you lots about its insides.

Another source for information is a magazine for CoCo owners, the RAINBOW. Now almost one year old it has grown to some 60 pages in the latest issue, has several software ideas in each issue and ads from those outside folks who are writing some pretty neat software and making some rather clever, but at your own risk, hardware items. For information write to the RAINBOW, 5803 Timber Ridge Drive, Prospect, KY 40059.

As for us, we WILL continue to bring out more games in ROM (and some on cassette), and we WILL continue to advertise the CoCo as a home computer, BUT we will also bring out "serious" software and peripherals and books to support those of you who want to unlock some of the power within. We will not bring out a General Ledger; I think that's a little too much.

And for all of you who have sent me letters, thanks. I read them all, the CoCo product manager read them all, and all should have received an answer. You have given us many things to think about, and some of your requests will lead directly to additional products.

Until next month.



Radio Shack Network In The Classroom

Craig Moore
Computer Coordinator
Harriet Tubman Middle School
2508 N.E. Everett
Portland, Oregon 97232

The Radio Shack Model I and Model III microcomputers have a cassette port through which programs enter and exit the computer. Normally a cassette tape recorder is connected to the cassette port. When you enter "CLOAD", a program loads into the computer from the cassette tape recorder. When you enter "CSAVE", the program in memory exits the computer and is recorded on the cassette tape. If, however, one has several computers and a network controller, the cassette ports can be used to connect the computers so that programs can be transferred from computer to computer, rather than computer to cassette tape recorder.

The middle school computer-use class that I taught last year was very dependent upon Network I Controller and sixteen Model I 16K microcomputers. In this article I would like to share what I learned about the use of a network in the classroom.

(Editor's Note: These remarks also apply to the Network 2 System, which has the capabilities of Network I, but is faster.)

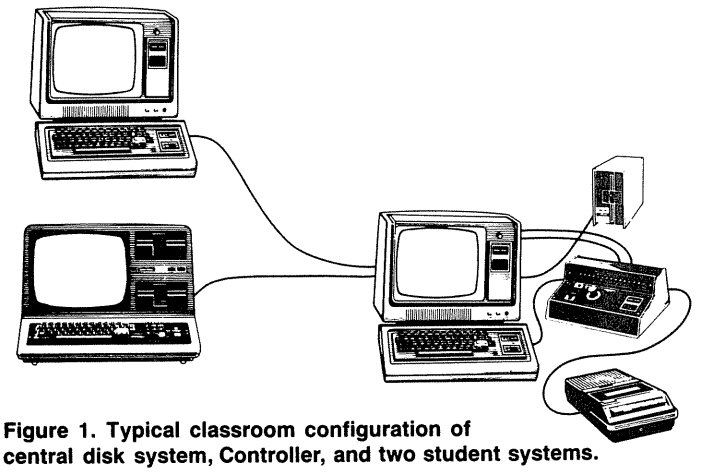


Figure 1. Typical classroom configuration of central disk system, Controller, and two student systems.

The Network I Controller is a controller (control box) and seventeen single strand cables (see fig. 1). One cable connects the controller to a computer with disk system that is designated as the "host". The remaining sixteen cables connect the controller to as many as sixteen microcomputers, referred to as "student stations". This equipment was placed on a rectangular arrangement of tables that occupied 3/4 of the floor space in my classroom (see fig. 2).

One use of the network is to send a program currently in the host computer, through the controller to the student stations. When the host operator enters the command "CSAVE", any student station with the command "CLOAD" will receive the program issued by the host computer.

This is a common procedure in my class for "handing back" student work. I keep student written programs from each class on a separate diskette. When students come to class they write the name of their program on the board. The host operator (often a student) inserts the class diskette and loads a student program into the host computer. The student enters "CLOAD" at the student station. A light comes on the controller indicating the student station is ready to receive the program. The host operator sees this, enters "CSAVE" and the program is transferred to the student station. If all sixteen controller lights were on, all sixteen student stations would receive the program. Students often will enter "CLOAD", before it is their turn, to view other student's programming efforts.

Because the program is transferred at the same 500 BAUD rate as in a cassette loading operation, the handing out of work can take as long as ten minutes (the time varies depending on the length of the programs that are sent). (Editor: Using a Network 2 Controller and TRS-80 Model III host and student stations, you can do this at 1500 BAUD—three times faster.)

The network can be used to send students instructional programs. For example, to illustrate the flow of activity in nested FOR-NEXT loops, I prepared a demonstration program. At the proper moment in my lecture, I had the students CLOAD the program from the host. While they ran the demonstration program, I asked them questions to focus their understanding of the nested FOR-NEXT loops.

Our schools scheduled one period each day to use the network for math computer assisted instruction. The programs we used (Radio Shack K-8 Math) had a wide range of skill levels and accommodated our ability grouped student's need for individualized lessons. It was possible to serve two groups, of sixteen students each, in the 45 minute class period.

Without a network controller, each student station would require a cassette tape recorder for saving programs. Although each TRS-80 Model I microcomputer system comes complete with a cassette recorder, I was relieved to not have the responsibility of accounting for sixteen recorders each period. With the network, a student would tell the host operator he/she wanted to save a program. The host operator

would turn a dial on the controller to the number of the student station and enter "CLOAD" on the host computer. When "CSAVE" was entered at the student station the program would transfer to the host. The host operator would then save the program on the class diskette or send it to the line printer for a hardcopy of the program.

Programs can also be sent from student station to student station with the same process as student station to host. Any computer with a "CLOAD" will pick up the transfer.

The handing in of programs at the end of a class session could take up to ten minutes.

The drawback to the use of a network controller is the slow transfer of programs from computer to computer. (Editor: This problem is reduced if you are using the faster Network 2.) If, in CAI class, the teacher has to load several long programs the time on task is shortened for the students whose program is loaded last. The teacher of a programming class may spend 20 minutes of class time transferring programs (although by saving programs in the same order as they were loaded, students should not have to wait more than 1/2 the total transfer time).

If a network were not used to transfer student programs to a central disk system for storage, but rather each computer had its own means of storage, i.e. cassette recorder, there would be no transfer time but the review of student work would be much more difficult. The teacher would have to pull programs off many different cassettes.

With the network, when student programs were to be graded, I would put the class diskette into the drive of the host system and load a student program (loading off the diskette is much faster than loading off a cassette). Then I would make a listing on the printer and add notes about how the program ran. I was usually able to view and make listings of an entire class in a 45 minute period. Corrections and suggestions were usually added later when the listing was reviewed in detail.

We have begun the 1981-1982 school year with an additional seventeen-computer network system. Unlike our first system, the computers are TRS-80 Model III and the network controller is the improved Network 2. The advantages to this system include simplified operation and, most importantly, a 1500 Baud rate. This means that the program transfer time is significantly reduced.

Clustering many computers in one classroom has definite advantages. The high access to the computer that is gained by clustering allows programming students to learn a great deal by demonstrations and trial and error. The network facilitates the clustering of many computers in one room by making possible rapid, organized and inexpensive storage and retrieval of programs.

A computer network can provide computer assisted instruction to a whole class. Given a good collection of educational software, all teachers in a building should be able to schedule their class for CAI. The quicker transfer rate of the Network 2 suggests the possibility of a teacher loading several different programs to different groups within the CAI class, without a significant loss of time at task.

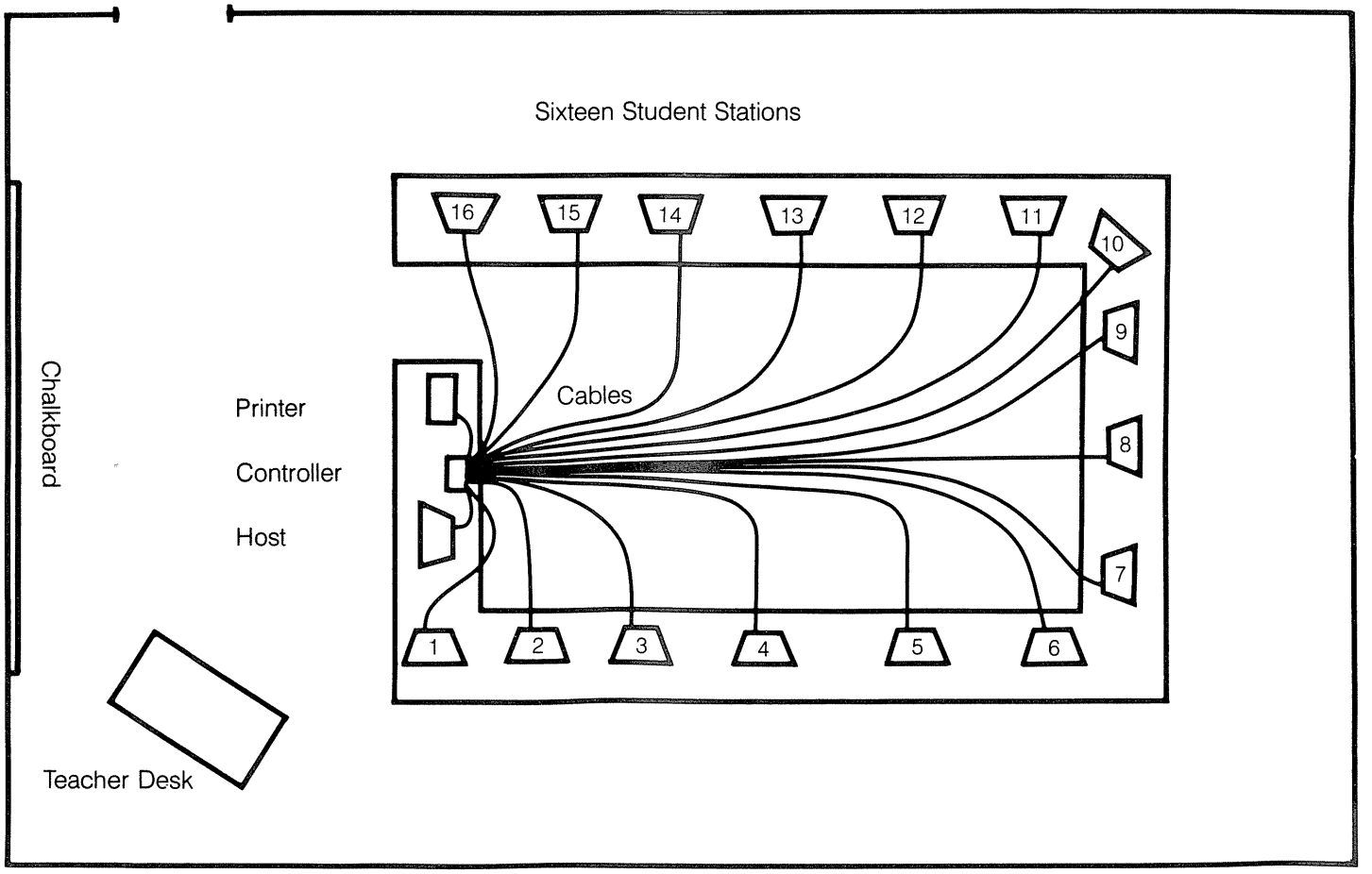


Figure 2. A Typical Classroom Setup

Search and Rescue and TRS-80

By Jerry Wellman,
840 E. 6th Ave.,
SLC, UT 84103

Can a TRS-80 have an important role in a search and rescue mission? Definitely, say several Utah TRS-80 users as they have demonstrated the use of the machine as members of Utah Civil Air Patrol.

(Civil Air Patrol is a nationwide, volunteer organization dedicated to search and rescue, aerospace education and a youth (cadet) program. National Headquarters of the CAP can be contacted at: Civil Air Patrol, Maxwell AFB, Alabama 36112.)

Some of the functional programs include compiling search effectiveness reports, listing a day's aircraft search effectiveness and -- get this -- offer a computer recommendation of best allocation of air search planes for the next day's effort.

Other programs keep track of up to 50 planes participating on a search (listing pilot information, time launched, time landed and total hours flown), and programs that convert latitude and longitude to the Air Force standard search grid (based on aircraft sectional maps) or converting the search grid back to latitude and longitude.

There are programs that teach search and rescue technique by way of game format and programs that keep track of people participating in search missions.

One of the most useful items of software is the "Aircraft Data" program that keeps track of Utah Wing CAP's aircraft. This program shows the status of the aircraft (including member-owned planes), the color of each plane, and other items involved in search missions concerning pilot qualifications, plane equipment and configuration.

The air data program runs on TRS-80 Model I, 32K, with printer and disk. Other programs will run on Level II, 16K.

Because TRS-80 and Civil Air Patrol are "nationwide" there is no reason the two cannot combine to improve search and rescue (SAR) effectiveness. (The search programs are easily changed to meet local requirements or even those of other groups apart from CAP involved in SAR missions.)

TRS-80 owners who would like to obtain the software with the intent of assisting CAP or another search group are invited to contact Lt. Col. Jerry Wellman, Utah Wing Civil Air Patrol, 840 East 6th Avenue, Salt Lake City, Utah 84103.

There is NO CHARGE for putting the search software on a tape or diskette. Software listings are available as time permits printing them out. Interested persons in the software should write first to make arrangements for copying. Readers who might be interested in Civil Air Patrol membership who are pilots, amateur radio operators, youth, etc., should write to CAP National Headquarters. Lt. Col. Wellman's offer is for software only (and he is interested in any related programs other CAP member-TRS-80 owner might have to share).

Figuring an I.R.A. Account

Emanuel C. Hoover
Rural Route 2 Box 172
Star City, IN 46985

This program figures I.R.A. type accounts at any amount of years and interest. The program is yearly compounded and puts results to a printer. It prints a set for the year span entered and the interest rate from the one entered down to six percent. After doing this, it asks for a new set of inputs. That is the reason for setting values to zero in line ten. Without that the values for all but the three inputs carry over into the beginning of the next set.

I am using this program with a 16K Color Computer and Line Printer VIII. We farm and have a small contracting business. We find this combination works fine for keeping farm records, figuring and printing grade survey figures, and pricing and billing materials. Our most recent purchase was the Spectaculator program, and we recommend it to anyone using a Color Computer for record keeping, inventory, or anything dealing with numbers.

```
1Ø CLS
    : A=Ø
    : B=Ø
    : C=Ø
    : D=Ø
    : E=Ø
    : F=Ø
    : H=Ø
2Ø INPUT"AMOUNT SAVED PER YEAR";A
3Ø INPUT"INTEREST RATE % PER YEAR";H
4Ø INPUT"NUMBER OF YEARS";C
5Ø B=H*.Ø1
    : B=B+1
6Ø D=D+1
7Ø E=A*(B^D)
8Ø F=F+E
9Ø IF D=C GOTO 11Ø
1ØØ GOTO 6Ø
11Ø PRINT""
12Ø PRINT""
13Ø PRINT"AMOUNT SAVED WITH INTEREST IS"
14Ø PRINT USING "##########.##";F
15Ø F=(INT(F*1ØØ))/1ØØ
    : A=(INT(A*1ØØ))/1ØØ
17Ø PRINT#-2,"$ ";A;" SAVED PER YEAR FOR ";C;" YEARS
    AT ";H;" %=$";F
18Ø D=Ø
    : B=Ø
    : F=Ø
    : H=H-1
19Ø IF H<6 GOTO 1Ø
2ØØ GOTO 5Ø
```

Computers in Education

The Computer Age is here! Teachers and parents are outnumbered by children who use computers, learn via computers, and accept computers as a part of their world. Computers have become a part of our society and everyday life patterns. Teachers and parents who know nothing about computers often feel PANIC as their children advance into this new age of computer literacy. We are confronted with a computer generation which is forcing traditional education down new paths of learning. This article is intended to reassure inexperienced adults that they can cope and even excel in computer literacy along with the younger generation.

Teachers and parents don't need a lot of knowledge to keep up with this new age. The key philosophy to be remembered is not to be afraid of these electronic devices. Teachers and parents are generally apprehensive about computers, but computers don't break easily. Change causes us to think, to adjust, and to grow. It is time for educators and parents to become aggressive and use this computer age to their benefit. Some of the following suggestions will help you become more familiar with computers and allow you to incorporate them into your life.

Building a good software library will allow a computer to become a good learning center in the school or at home. A software library may be developed in many ways: buying pre-existing software which is ready to run on your computer, keying in programs from numerous books that you will find for sale in computer stores (some computer literacy helps in debugging and correcting book programs), or writing your own programs for specific uses.

Many of the Radio Shack education programs are excellent in school settings and very helpful for home reinforcement in learning necessary skills. Before attempting to make use of these or other pre-existing programs, a teacher or parent should review the manuals which come with their computer. By the time you have familiarized yourself with your machine and how to hook it up with all of the peripherals, you are ready to learn the fundamentals of BASIC and what capabilities your computer has. TRS-80 Model III's and Color Computers come equipped with an Owner's Manual and a book called GETTING STARTED WITH TRS-80 BASIC. You will talk to your computer by typing on the keyboard and it will talk to you on the video screen. If the computer doesn't understand you, it will say so. You really can't break or damage your computer by typing on the keyboard.

After completing your GETTING STARTED WITH TRS-80 BASIC, you will feel more confident in working with programs to be typed in and in using pre-existing programs.

If you choose to try pre-existing programs, you will find a wide variety of programs with many different skills. Radio Shack has developed a special kind of computer program which is designed specifically for teaching called CAI (Computer Assisted Instruction). This method is based on the use of the computer as a medium of instruction. Our CAI Courseware development has added an additional instructional strategy. Our courseware products are developed and field tested, taking into consideration the feedback from teachers and curriculum developers. The CAI techniques which have proven most effective for supplementing the regular classroom instruction have been centered on individualized sessions where the computer generates and presents exercises for certain subject areas.

Many studies are being conducted on the impact of computers on education. Not only are they tools for learning, but they are a means for helping prepare students to live in a world increasingly dependent on computers. Radio Shack is meeting these education needs with their COMPUTER EDUCATION SERIES (CES). The Series is designed for use in the classroom at secondary and college levels. A teacher's manual, 25 workbooks, and a set of transparencies are included. PART 1: INTRODUCTION TO BASIC (26-2150) is designed as a first introduction to computer programming. PART 2: BASIC PROGRAMMING (26-2152) continues the concepts introduced in PART 1. PART 3: ADVANCED BASIC (26-2154) deals with more advanced programming concepts. These instructional materials require little programming or computer knowledge on the part of the instructor. All programming exercises are designed to be entered and run by students with hands-on access to the TRS-80. All three parts are designed for use with any Model I or Model III.

COMPUTER DISCOVERY is a new addition to our Computer Education Series. It introduces students to computers, their evolution and impact on society, and basic programming concepts. Each package includes interactive exercises, 25 student workbooks, and an instructor's manual. The two levels available are COMPUTER DISCOVERY FOR JUNIOR HIGH (26-2630) and COMPUTER DISCOVERY FOR SENIOR HIGH (26-2632). Additional workbooks are available separately.

Other pre-packaged courses designed to teach Basic Instruction include Model III BASIC COURSE (26-2010)—eight self-paced lessons on diskette to teach programming, Model III BASIC COURSE—TAPE (26-2015), and Model III DISK COURSE (26-2014)—four self-paced lessons on diskette that cover DISK BASIC and TRSDOS.

The TRS-80 MICROCOMPUTER INFORMATION HANDBOOK FOR EDUCATORS (26-2757) is a booklet which describes microcomputers in detail. What they are and how they can be used in classrooms is discussed along with costs, courseware, and other considerations which could be of special concern to the educator.

Another Resource Material for educators is BASIC COMPUTER LITERACY: COMPUTERS PAST AND PRESENT (26-2755). This classroom package includes a teacher's guide and six spirit masters illustrating the history of computers, what a computer is, what it can do, and how it works. This is an introductory computer education package

for grades four through eight. It explores the history of computers from the abacus and the first mechanical calculator to the programmable computers of today. A sample computer program is shown and analyzed. A colorful wall chart is also included. Additional activities and answers are printed on the non-duplicating side of the masters. This package was designed for effective use whether or not the class has access to a computer. It is not necessary for the teacher to have any previous knowledge of computers.

The PROPOSAL WRITING GUIDE (26-2754) is a step-by-step guide for educators seeking funding from an external source (such as the Federal government or a private foundation) for computer-related education projects. The guide leads you through procedures necessary to the writing of a successful proposal; detailing the most important elements.

MY TRS-80 LIKES ME (26-2751) is a new 32-page teachers' guide to helping elementary students understand and enjoy the BASIC computer language. It includes lots of programs and activity ideas.

NUMBER PATTERNS (26-2752) is a 36-page booklet showing how to introduce learners to sequences and series in an entertaining way using computers.

The TRS-80 EDUCATIONAL SOFTWARE SOURCEBOOK, VOLUME I (26-2756) has been introduced as a special edition of the TRS-80 SOFTWARE SOURCEBOOK for educational courseware. Vendor-furnished listings include a description of the program and important characteristics. It is indexed by subject, user level, and instructional technique.

A further aid for administrative personnel, available in the future, is a REKORD Planner which is a record-keeping system that can be personalized to fit your own needs. It will file and store student data, print out student and school records, and generate reports. It will be available in three versions: Administrator's version (26-2725), Counselor's version (26-2726), and Special Programs version (26-2727). It will be available for use on the Model III.

Our Educational Software includes a variety of Math programs ranging from K-8 MATH for the elementary grades through ESSENTIAL MATH for grades 7 through 12. K-8 MATH has a Student Management version which assists the teacher by keeping a complete record of all lessons and tests taken by each student. This feature allows the teacher to review and print out student reports. It automatically keeps track of lesson promotions and demotions sending each student to the correct lesson at the beginning of a new student session. The K-8 MATH, VOLUME ONE (26-1715) is for use with TRS-80 Model I or Model III. It is furnished on cassette and diskette and requires a minimum system of 16K tape or 32K 1-drive. The K-8 MATH WITH STUDENT MANAGEMENT PROGRAM, VOLUME ONE (26-1725) is for use with Model I or Model III. It is furnished on diskette only and requires a minimum system of 32K 1-drive with a line printer being optional. Unless otherwise indicated, all of the education programs are for use with TRS-80 Model I (with Level II BASIC) or Model III (with Model III BASIC).

The K-8 MATH WORKSHEET GENERATOR program provides an added feature in allowing a teacher to design and print worksheets and answer sheets for the addition, subtraction, multiplication and division programs of the K-8 MATH program. A teacher can design headings for the worksheet, choose the lesson numbers and the number of problems from each lesson, print identical or different worksheets on the same level of difficulty and then save the worksheet design on diskette for future use. K-8 MATH WORKSHEET GENERATOR (26-2162) is for use with Model I or Model III. It is furnished on diskette and requires a minimum system of 32K 1-drive with a line printer.

Included with each of these programs is the Radio Shack K-8 MATH CROSS-REFERENCE (26-2750) which coordinates individual K-8 MATH lessons with specific chapters and pages in six widely used math basals.

The ESSENTIAL MATH PROGRAM, for grades 7 through 12, reinforces math concepts which have been introduced by the teacher. VOLUME ONE provides skill building exercises in addition, subtraction, multiplication, division, and number concepts. VOLUME TWO covers fractions, decimals, percents, and pre-algebra concepts. This program also includes a placement mode, as well as an option that allows automatic promotion and demotion of the student through the lessons. Constant feedback is provided to the student with reinforcement messages keyed for correct and incorrect responses. At the end of each lesson, a report giving the student's score appears on the screen. Scores can then be recorded on copies of a sample student record sheet included in the teacher's manual. This will provide a complete record of each student's progress. ESSENTIAL MATH PROGRAM, VOLUME ONE (26-1716) and ESSENTIAL MATH PROGRAM, VOLUME TWO (26-1719) are for use on Model I or Model III. It is furnished on cassette and diskette and requires a minimum system of 16K tape or 32K 1-drive.

The EUCLID GEOMETRY TUTOR program is designed to reinforce basic geometry concepts introduced by the teacher. It allows students to develop proofs using nine basic postulates of Euclidean Geometry. A teacher's manual includes an exercise section with a wide variety of problems. A quick-reference card is also included for use with the program. The EUCLID GEOMETRY TUTOR (26-1724) is for use with Model I or Model III. It is furnished on cassette and diskette requiring a minimum system of 16K tape or 32K 1-drive.

A series of Math and Physics Programs for the Secondary Grades are available for simulation and problem solving. All of these programs are for use with the Model I or Model III. They are furnished on cassette and diskette and require a minimum system of 16K tape or 32K 1-drive. The following programs are included:

ADVANCED GRAPHICS (26-1714)
VECTOR ADDITION (26-1720)
INTERPRETING GRAPHS IN PHYSICS (26-1721)
GRAPHICAL ANALYSIS OF EXPERIMENTAL DATA (26-1722)
INVESTIGATIONS IN INTEGRAL CALCULUS (26-2600)
PLANE ANALYTIC GEOMETRY (26-2602)

NUMERIC DATA ENTRY PRACTICE is a 25-lesson practice course designed to help students develop speed and accuracy in the 10-key entry of numeric data. Students use the TRS-80 standard numeric keypad to "key in" numbers that appear on the video screen. Students work through exercises of progressive difficulty. The computer records the student's performance on diskette for review by the teacher at a later date. Student reports may also be printed out using a line printer. NUMERIC DATA ENTRY PRACTICE (26-2601) is for use on the TRS-80 Model I or Model III. It is furnished on diskette and requires a minimum system of 32K 1-drive with a

line printer being optional.

More and more teachers are becoming aware of the instructional potential of microcomputers. An area of increasing interest to many teachers is that of reading instruction using the computer.

One of our first programs in that area is AlphaKey™, which is an alphabet and keyboard familiarization for children four to six years old. Parts 1 and 2 can teach children to discriminate capital letters, to associate capital and lower case letters, and to recognize the order of the alphabet. Part 3 gives the child practice in typing the letters of the alphabet in order with no hints. Immediate reinforcement is given for every response. A Happy Face appears for correct answers. If an incorrect answer is typed, the Happy Face will disappear until the correct answer is given. A report is given at the end of each part showing letters identified correctly on each try along with average response time. AlphaKey™ (26-1718) is for use with Model I or Model III. It is furnished on cassette and diskette requiring a minimum system of 16K tape or 32K 1-drive.

The HIGH MOTIVATION READING SERIES is designed to interest and challenge students at reading levels 4-6. Each package in the series contains four copies of an illustrated-format reader, and a read-along audio cassette tape. Educational activities related to the story in the reader are furnished on diskette for students to work on at the computer. Additional student activities are included in the teacher's manual. The HMRS STUDENT RECORDS SYSTEM (26-2521) is available separately and can be used with any of the Model III HIGH MOTIVATION READING SERIES programs to maintain a complete record of student activity scores. Titles in the series include:

CHARLES LINDBERGH/AMELIA EARHART (26-2513)
THE HOUND OF THE BASKERVILLES (26-2514)
DRACULA (26-2515)
MOBY DICK (26-2516)
THE BEATLES (26-2517)
20,000 LEAGUES UNDER THE SEA (26-2518)

The HMRS™ is for use with the TRS-80 Model I with lower-case modification and the Model III. It is on diskette and the minimum system requires the use of the TRS-80 MicroPilot™ program in conjunction with all seven titles. The HMRS STUDENT RECORDS SYSTEM does require a 32K 2-drive system with a line printer being optional. All of the other HMRS programs require 32K 1-drive systems. All of the catalog numbers listed above for the HMRS are for the Model III versions. Please check with your local Radio Shack dealer for availability of these programs for the Model I.

Radio Shack also has four READING IS FUN packages available for the Color Computer. They are designed for grades 4-6. Each package includes an illustrated reader, a read-along audio cassette tape, and a computer tape with spelling and vocabulary exercises. DRACULA (26-2550), THE HOUND OF THE BASKERVILLES (26-2551), MOBY DICK (26-2552), and 20,000 LEAGUES UNDER THE SEA (26-2553) require a 4K Color Computer and a CTR-80A cassette recorder.

The Radio Shack COMPUTER ASSISTED READING DEVELOPMENT program is adapted from the Philadelphia Reading Program and was designed to help develop the reading skills of students at the upper elementary level. C.A.R.D. 1: SENTENCES provides various exercises and tutorials in recognizing sentences and sentence parts. Other modules will include PARAGRAPHS, DIRECTIONS, and COMPREHENSION. These programs are for use with TRS-80 AUTHOR I or the TRS-80 AUTHOR I LESSON PRESENTATION PACKAGE. The C.A.R.D. modules include both the lesson program and a student record keeping feature. C.A.R.D. 1: SENTENCES (26-2603) is for use with the TRS-80 Model III and is furnished on diskette. It requires a minimum system of 32K 1-drive and also requires TRS-80 AUTHOR I or the TRS-80 AUTHOR I LESSON PRESENTATION PACKAGE.

Another group of programs designed for Educational use include the authoring systems. These programs allow teachers and users to create their own computer assisted instruction material.

The QUICK QUIZ program allows teachers to create, store, and give multiple-choice tests using a TRS-80 computer. It handles up to 40 questions with 4 answer choices per question. Up to 50 students can be tested and their scores stored on a disk file for later review. QUICK QUIZ (26-1728) requires a 32K Model I or Model III disk system.

TRS-80 MicroPilot allows the user to create or adapt courseware from curriculum suitable for computer assisted instruction to the TRS-80. It is a command-oriented author language and is based on the PILOT computer language. It has extended capabilities for graphics generation and file handling. A basic knowledge of computer programming is not required, but would be helpful. The MicroPilot package includes a sample lesson on the program diskette, a user's reference card, a screen reference card to aid in producing graphics, and a teacher's manual with complete instructions. MicroPilot (26-2205) is for use with the TRS-80 Model I and MicroPilot (26-2718) is for use with the TRS-80 Model III. It is furnished on diskette and requires a minimum system of 32K 1-drive.

TRS-80 AUTHOR I is a lesson development system designed to allow creation of courseware. Knowledge of computer programming is not necessary. A step-by-step instruction manual guides the user through all phases of lesson creation. The system allows text and graphics presentation, extensive branching, control over the number of tries allowed the student on a question, student access to glossary words provided by the author, hints to help students, and includes optional student record keeping. TRS-80 AUTHOR I (26-1727) is for use with the TRS-80 Model I or Model III and is furnished on diskette. It requires a minimum system of 32K 1-drive.

The TRS-80 AUTHOR I LESSON PRESENTATION PACKAGE is the portion of AUTHOR I which presents the lessons to the student using the TRS-80 computer. This module is available separately to allow teachers who do not want to develop their own lessons to use AUTHOR lessons created and sold by others. The TEACH lesson presentation program and the STUDENT record keeping program are provided with this module. The TRS-80 AUTHOR I LESSON PRESENTATION PACKAGE (26-2707) is for use on Model I and Model III and is furnished on diskette. It requires a minimum system of 32K 1-drive.

Recent additions to Radio Shack's line of Education Software includes HISTORY OF TECHNOLOGY for the Color Computer. PIONEERS IN TECHNOLOGY (26-2624) includes The Age of Flight, Space Exploration, The Electric Car,

and History of Computers. INVENTIONS THAT CHANGED OUR LIVES (26-2625) includes Edison's Electric Inventions, Bell and the Telephone, The Story of Railroads, and The Age of Television. These packages describe technological achievements through graphics, text, and recorded speech. A
16K Color Computer and a CTR-80A cassette recorder are required.

VOCABULARY TUTOR I (26-2568) and VOCABULARY TUTOR 2 (26-2569) are future programs which provide a fun way for your child to learn new words. They use the new Color Tutor system which combines pictures and text on the screen with spoken information and feedback.

An additional program for the Color Computer is TRS-80 Chemistry Lab, Volume I. The program includes a set of chemistry laboratory experiment simulations which allow students to control and see the results of various experiments many times without the expense or danger of using chemicals and labware. Titles in Volume 1 include Kinetic Theory, Charles' Law, Boyle's Law, Titration, Conductivity, and Solubility. The package includes an instruction manual and student experiment books. Volume 2 of Chemistry and two volumes of physics simulation are planned. TRS-80 Chemistry Lab, Volume I (26-2609) requires a 16K Color Computer with Extended BASIC and a CTR-80A cassette recorder.

The Radio Shack Education Division has also made a commitment to the hardware portion of student instruction. The TRS-80 NETWORK 2 CONTROLLER and the NETWORK 3 CONTROLLER allow an educator to design a network of up to 16 non-disk student stations around a single disk "host" system. With the NETWORK 2 CONTROLLER, a teacher can download programs into all 16 student stations at once, or into any combination of the student stations. Different programs can be loaded into different stations. Student programs can also be uploaded to the host. The NETWORK 2 (26-1211) is for use with a host computer that can be either a Model I or III (with at least one disk drive recommended) or a Color Computer (with a cassette recorder or disk drives). From one to sixteen student stations can be used. The TRS-80 Systems used as student stations do not require disk drives or cassette recorders. Model I and Model III computers can be mixed in a NETWORK 2 configuration. If Color Computers are used with a NETWORK 2, all computers in that NETWORK (including the host) must be Color Computers.

The NETWORK 3 (26-1212) is designed to allow nondisk student stations to access the disk drives and printer of the teacher's host system. This provides the student stations with most of the same disk features that make a disk-equipped computer so powerful.

Some of the features of the NETWORK 3 CONTROLLER are:

* All 16 student stations can load and save programs using the disk at the host system, as well as use a line printer attached to the host system - without interrupting other students or the teacher.

* Student stations can store information on the host disk, allowing class records to be kept in one location.

* Each student station can be a part of the Network, or operate independently without disturbing the network cable connections.

* Teachers can use the included MENU program to create a "menu" (or option list) of programs for students, allowing them to choose from a selection of programs on the disk at the host computer.

* NETWORK 3 makes it possible for computer science students to learn and use many of the Disk BASIC functions on non-disk TRS-80s.

The NETWORK 3 is designed to operate best in a student-centered classroom, a lab, or a resource room situation. It requires no teacher intervention under normal operating conditions. These features complement the NETWORK 2, which operates best in a teacher-oriented environment and requires teacher coordination of students and computers.

An operating system, "STUDENT", must be loaded into each student station via the SYSTEM command from cassette tape or NETWORK 1 or NETWORK 2 before NETWORK 3 operation begins. A planned ROM initializer will eliminate this cassette load and allow the student station operating software to be loaded directly from the Host disk. When the ROM is available, it can be installed at any Radio Shack Repair Center. (Contact your local Radio Shack Store or Computer Center for details.)

Student stations will need to have adequate memory for NETWORK 3 operating software and student application programs. In a 16K student station, STUDENT and NBASIC programs occupy about 9K of memory, leaving 7K. If more than 7K of student application program space is needed, additional RAM may be purchased in 16K increments up to a maximum of 48K.

The NETWORK 3 equipment required for this system set-up includes the NETWORK 3 CONTROLLER (26-1212), the NETWORK 3 Operating System (26-2775), the appropriate length RS-232 Cable for each computer, and a 6" RS-232 Card Extender (26-1497) for each computer. Also included in this set-up would be the host and student station computers and installation of RS-232 interface boards in all computers.

Many Radio Shack Software programs are fully compatible with the Network 3. Radio Shack educational software packages compatible with the initial version of the Network 3 Operating System include:

K-8 MATH PROGRAM
K-8 MATH WITH STUDENT MANAGEMENT
ESSENTIAL MATH, VOLUMES ONE AND TWO
EUCLID GEOMETRY TUTOR
ADVANCED GRAPHICS
VECTOR ADDITION
GRAPHICAL ANALYSIS OF EXPERIMENTAL DATA
INTERPRETING GRAPHS IN PHYSICS
INVESTIGATIONS IN INTEGRAL CALCULUS
PLANE ANALYTIC GEOMETRY
AlphaKey

Since the NETWORK 3 is for use in an educational setting, it does not support the Radio Shack Business Programs. The major reason for this is that only one student station may access and open a particular file at one time, which prevents all other student stations from accessing that file until it is closed. Different student stations may access different files at the same time. A special version of SCRIPSIT for Education and a Business Education Series is being planned which will be compatible with the NETWORK 3.

Radio Shack offers educators a variety of different services. Our National Bid Department handles the school bids and offers the best possible price quotations on quantity

purchases of TRS-80 equipment and courseware.

Our National Lease Department is able to put the TRS-80 into classrooms with terms to meet almost any budget. Because you can lease everything at once, instead of buying equipment gradually, you can provide more hands-on experience for the students immediately.

The operation of a TRS-80 requires more training than just learning to operate a movie projector or a tape recorder. Since some instruction and training is desirable, Radio Shack offers free training sessions to teachers and school administrators. Our instructors are able to conduct these classes to fit your schedule. Contact your nearest Radio Shack Computer Center for details.

Most of our Radio Shack Computer Centers have a Customer Service Representative to assist customers with any problems which may develop. If you desire more help than your store is able to provide, Computer Customer Service has an Educational Software group (817-390-3302) for helping educators and users of the educational packages. These people are able to help you implement a computer learning program, answer questions on new products, and also help to apply packages to your individual classroom. The Educational Software group will help talk you through any problems with existing programs. However, they are not able to help you modify any of the existing software programs. They realize the importance of the educational use of computers and are dedicated to helping you get the most out of your TRS-80 classroom.

Accept the challenge of the Computer Age—become informed and learn with the new generation—use your Radio Shack Resources to help educate and equip you to deal with this age of advancement in computer technology.

Computer Customer Services Address and Phone Number

8AM to 5PM Central Time

Computer Customer Services
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

These are New Numbers and Were Effective Aug. 27

Model I/III Business Group	817-870-2041
Model II/16 Business Group	817-870-2042
Languages and Compilers	817-870-2044
Color and Pocket Computer Group	817-870-2150
Hardware and Communications Group	817-870-2571
Educational Software	817-390-3302
Games, Books, and New Products	817-870-2271

Figure It Out With Alice and Mac

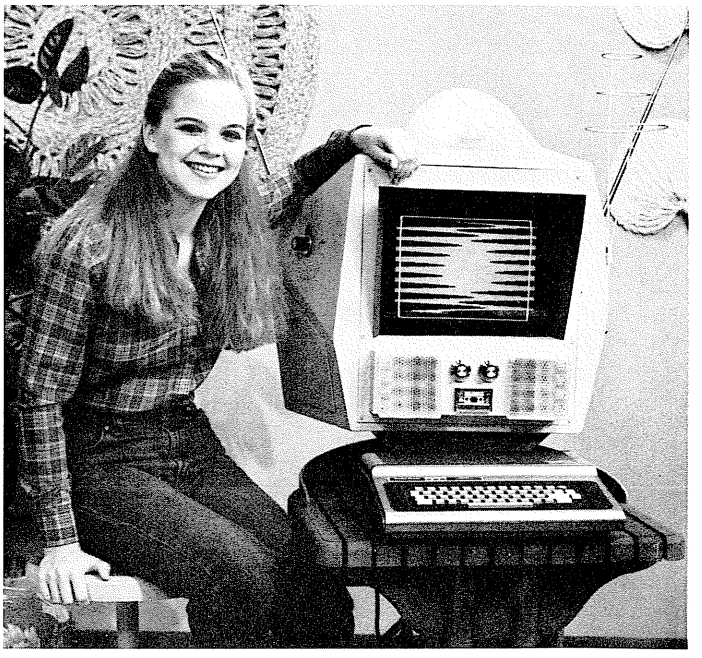
The TRS-80 Color Computer will be seen on some Educational Television stations this fall in a new mathematics series called "Figure Out". Tandy Corporation donated the Color Computer to the station that produced the show—the Mississippi Authority for Educational Television.

"Figure Out" presents the adventures of Alice, a young 'computer whiz', and Alice's business partner Mac, a computer with a mind of his own and a lot of personality. The TRS-80 Color Computer is part of Mac's body, along with a large monitor decoratively enhanced by the art department at Mississippi ETV.

Alice and Mac work together in Alice's computer programming company (called "Alicesystems"), located in Alice's apartment in an old observatory. Some of their adventures involve trying to get the business going in spite of an villainous competitor who is trying to destroy Mac. Comic features of the show include Mac's attempts to flirt with Alice's refrigerator.

The math concepts presented are part of the story, emphasizing the practical applications of math. Alice and Mac figure out problems such as how much they owe the air conditioner repair man, how much money Mac will make if he goes to Hollywood to become a singing star, and what the shortest route is to a destination. Usually the characters figure out the problem, a narrator explains the concepts involved, and the viewer gets to solve practice problems. Problems are displayed and solved on the television screen using large graphic numerals, pencil and paper, and at times a calculator. Concepts from numeration and rounding to multiplication and division are covered.

At the same time as they entertain, the lessons cover a lot of math in a cumulative and systematic way. Four lessons are on numeration and rounding, one is on addition, three lessons each are on subtraction and multiplication, and division is covered in four lessons. Because the series emphasizes use of traditional tools to solve math problems, Mac's role as a character is at least as important in "Figure Out" as are his actual capabilities as a computer. But "Figure Out" makes the important point that computers don't need to be cold, frightening machines. With the right programming, a computer can be a warm and friendly partner or helper. Alice and Mac prove it because, as the narrator says, "together they can figure out anything"!



The Multiple Choice Puts You to the Test

Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer NEWS, various features of CompuServe will be discussed.

The CompuServe Information Service is sold at Radio Shack stores nationwide and in Canada.

You want to attend graduate school, and wonder how you'd fare on the entrance exam. Your child has mastered reading, and he's anxious to show off his news skills. Or, what if the world's most accomplished trivia buff has nothing on you, and you want to prove it?

The answers can be found in one of CompuServe's home services offerings — The Multiple Choice (TMC).

TMC, number eight on the Home Services menu, offers tests and games the entire family can enjoy together. It was designed that way by its creator, Howard Millman, a professor of educational psychology at the University of Maryland.

Millman had heard about CompuServe, and the idea of learning by computer intrigued him. He thought about the value of a program that would make learning fun and educational, and that families could enjoy together. He bounced some ideas off of friends and family, contacted CompuServe with a proposal, and TMC was born about a year ago.

Millman, with the help of family and friends, composes all TMC questions himself. A friend provides programming advice and helps him program TMC from his home.

A glance at the TMC menu shows plenty of choices. If it's pure fun you're after, play "Trivia Unlimited", which lets players compete against themselves and others by answering a wide variety of trivia questions. If your recall skills can stand a little brushing up, the 10-question warm up quiz is probably your best choice. But if you're the type who pores over little-known facts, the "sudden death" option is for you. Questions are much more difficult, and the scoring method truly tests your mettle: you keep answering questions until you've missed three; then the test ends and your score is compared with the top three scores by all other test takers.

The children can choose a TMC designed especially for them. "TMC for Kids!" has five tests and games that are fun and help perfect already-learned skills. "Silly Fill-Ins" asks kids seven and older to fill in missing words in a story. A sports quiz tests knowledge of basketball, baseball, soccer, football and tennis rules and terms. "J*U*M*B*L*E*D Words" asks players to recognize disguised words, while another game tests knowledge about the 50 states. And if that grade-schooler of yours knows what color Kermit's bicycle was in "The Muppet Movie", they'll like "Trivia for Kids".

In a more serious vein, Millman also created an intelligence quotient exam and one that provides some insights into who you are. "The TMC Intelligence Test", while designed to be similar to an actual IQ exam, is not meant to be a substitute for one. But the results "should give you a fair estimate of how smart you are", Millman maintained.

"The TMC Personality Profile" was created based on general personality patterns, Millman said. Questions cause the test taker to look at his psychological, physical and spiritual self, and measure how he views relationships and decision making. At the test's end, the computer provides a chart that shows how well the test taker is caring for himself in each of eight areas examined.

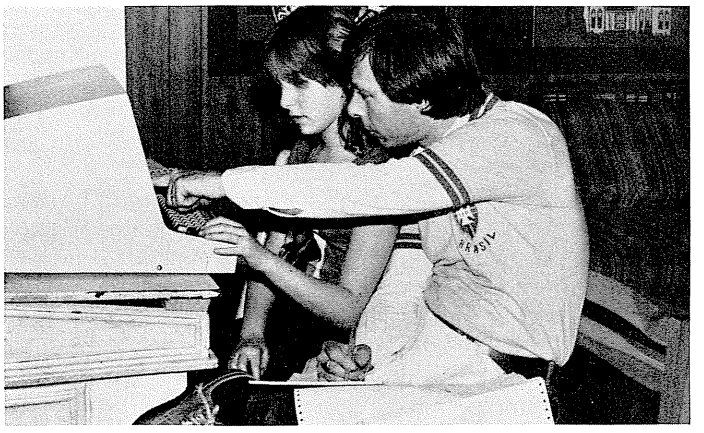
Perhaps the most serious TMC category is "TMC Analogies", which simulates a national exam taken by graduate school applicants. Questions are close to the test in length, type of analogies and trick questions. Test takers can choose a practice quiz or pick the more difficult timed exam.

After investigating any of the TMC categories, Millman invites users to tell him what they thought via "TMC Feedback". Millman personally reads all the "letters" he receives and answers many of them.

"I get mostly good comments", he said. "People give me lots of pep talks. They said 'thanks, I needed that', or 'keep up the good work.'"

Keeping up the good work is something Millman is always striving for. He's now in the process of programming 10 more TMC categories into the system, and a sneak preview reveals that there will be something for everyone to enjoy.

Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, editorial director, CompuServe Information Service, P.O. Box 20212, Columbus, Ohio 43220 or through Feedback on the User Information menu.



University of Maryland Professor Howard Millman reviews questions for the Multiple Choice on CompuServe with his daughter.

Window On Wall Street

The Corporate Earnings Estimator

Editor's Note: With over 45,000 subscribers, Dow Jones News/Retrieval is the leading provider of online business and financial information. We will keep you up-to-date on new software, new data bases and new prices in upcoming issues of TRS-80 Microcomputer News. Dow Jones News/Retrieval is sold at Radio Shack stores nationwide and in Canada.

Wouldn't you like to see the same earnings per share estimator that is available to professional investors before you decide to buy or sell a stock?

Try the Corporate Earnings Estimator. It is for those of you who do not want to rely on a single broker for investment advice, and who are aggressive enough to decide, on your own, how your investment portfolios should be structured.

The Corporate Earnings Estimator makes it possible for the first time in the history of the stock exchange, for the average investor to see the full range of Wall Street brokerage estimates.

Before the service was offered by News/Retrieval, only major institutions were able to make decisions based on earnings per share (eps) estimates that were as complete, up-to-date and accurate as possible. But now, you can see the same information yourself.

The Corporate Earnings Estimator provides a statistical summary of brokerage analyst eps from the top firms on Wall Street, which include 1,200 analysts at 50 major brokerage firms.

"We receive virtually all of the brokerage research published in the United States and we have a staff that goes through it every day", said Leonard Zacks, founder of Zacks Investment Research, Inc., which provides the service to News/Retrieval.

"You get all of the estimates in a statistical, summarized form . . . the mean, the high . . . the low," Zacks said. "Most individual investors' buy/sell decisions are made on incomplete information. By looking at the eps estimates, you can make better decisions to buy stock".

"We pull out the estimates from the brokers' reports. Then we send Dow Jones News/Retrieval tapes each week", he said, who in turn make the information available to News/Retrieval subscribers.

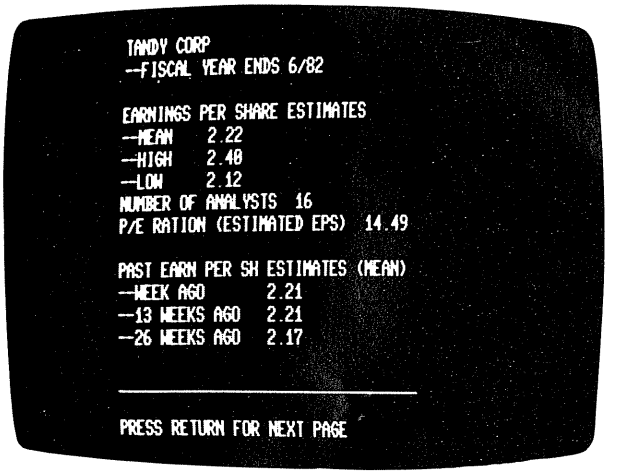
Zacks initially founded the company in Chicago in 1977 to assist institutional investors in using quantitative technology in their investment processes.

Before founding the company, which now provides brokerage analyst eps estimates, rankings and software systems, primarily to institutional investors, Zacks was Vice-President of A.G. Becker and Company, Inc., which was responsible for the design of quantitative research services, performance measurement services and econometric analyses.

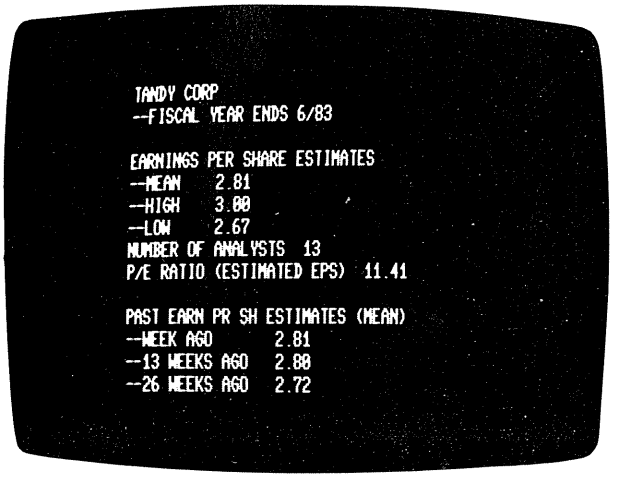
Zacks holds a Ph.D. from M.I.T. and is the author of several articles in the quantitative and financial areas.

To retrieve Corporate Earnings Estimates on the Dow Jones News/Retrieval service, just type: //EARN and press the ⟨ENTER⟩ key.

Once in the data base, access a particular company by typing the company's stock symbol (located in the Dow Jones News/Retrieval Operating Guide and Directory of Symbols), followed by ⟨ENTER⟩. To view a second page of estimates for the following fiscal year, just press the ⟨ENTER⟩ key again.



Screen 1



Screen 2

You can get about two stocks per minute when you access the service with a personal computer or terminal.

Peter Sohn, who works with the New York based money firm, Tweedy, Browne, Inc., said he uses the data base daily.

"It comes in as a second opinion", Sohn said of the service. "The information on the estimator gives you kind of a tip-off sometimes to something that someone sees and the other ones do not see . . . You get diversity and you get more information and it is accessible for the little guy".

Sohn said he checks three to five companies "at pop". "Both News/Retrieval and the estimator have achieved that ultimate distinction of becoming indispensable", he said.

Barney Ziv uses the estimator to retrieve earnings estimates on portfolios managed by his Chicago based investment firm.

"We are fairly new subscribers, but we like the service. It provides a service that we will probably use increasingly", he said.

Zacks said brokers do not see the service as a threat. In fact, he said they "love it," because they can get all of the information also.

With other services, Zacks said you do not get information that is particularly up-to-date. Sometimes, it can be as many as three months old, he said.

But with the Corporate Earnings Estimator, the information is updated daily and reports for News/Retrieval are generated weekly.

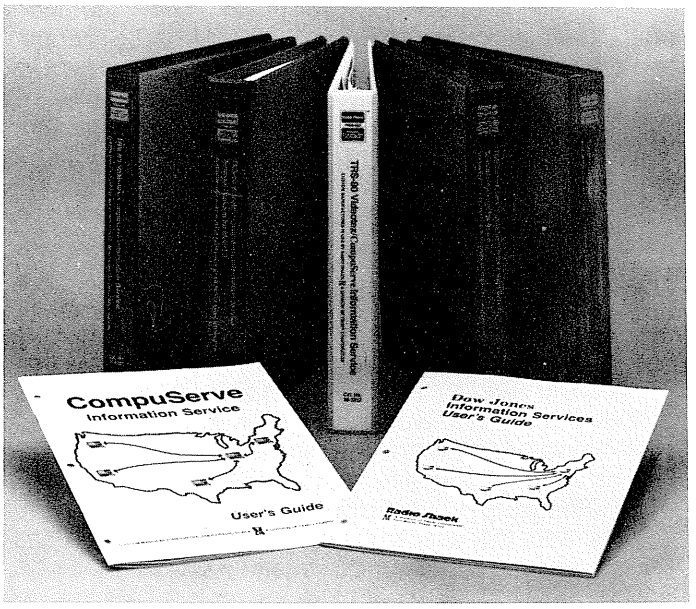
If you would like more information on Dow Jones News/Retrieval, or have any questions on the service, please call the Dow Jones Customer Service Hotline at:

1-800-257-5114

in New Jersey and Canada Call:

609-452-1511.

DOW JONES NEWS/RETRIEVAL®



microRESEARCH Problems

Richard V. Andree
microRESEARCH
P.O. Box 2910
Fort Worth, Texas 76113-2910

Before we go on with microRESEARCH problem #2, here is a brute force program to solve problem #1:

```
1Ø CLS
2Ø FIRST=SQR(1Ø23456789)
3Ø FIRST=INT(FIRST)
4Ø LAST=SQR(9876543210)
5Ø LAST=INT(LAST)
6Ø FOR I=FIRST TO LAST
65 FLAG$="YES"
7Ø TEMP#=I*I
75 'PRINT@Ø,TEMP#
8Ø TEMP$=STR$(TEMP#)
9Ø TEMP$=RIGHT$(TEMP$,LEN(TEMP$)-1)
1ØØ IF LEN(TEMP$)<1Ø THEN 999
11Ø IF LEN(TEMP$)>1Ø THEN I=LAST
     : GOTO999
12Ø FOR J=Ø TO 9
122 F(J)=Ø
123 NEXT J
125 FOR J=1 TO 1Ø
13Ø J$=MID$(TEMP$,J,1)
14Ø K=VAL(J$)
15Ø F(K)=F(K)+1
16Ø IF F(K)>1 THEN FLAG$="NO": J=1Ø
17Ø NEXT J
18Ø IF FLAG$="NO" THEN 999
19Ø PRINT C,I, TEMP#
2ØØ C=C+1
999 NEXT I
```

This program was written on a TRS-80 Model II by Microcomputer News Editor Bruce Elliott, and will list the integers whose squares contain each of the ten digits (0-9) exactly once. I call it a "brute force" program, because there is nothing elegant about it. The program simply starts with appropriate values for FIRST and LAST, that is the smallest and the largest possible integers that might yield a proper result, and then uses a FOR-NEXT loop to examine every value in-between. How does this compare with your solution? If I get a more elegant solution in the next month or two, I will publish it.

microRESEARCH problem #2.

A **palindrome** is a phrase or sentence which reads the same forward as backward if you disregard spaces and punctuation. Example: "Ma is as selfless as I am."

There are also number palindromes. It is fairly easy to find perfect squares which are palindromes.

$(202)^2 = 40804$: $(264)^2 = 69696$. There are infinitely many palindrome squares. However, palindrome squares having an **even** number of digits, such as: $(836)^2 = 698896$ are fairly rare. Only three palindrome squares having an **even number of digits** are known today. All three are discoverable using your TRS-80. Do so.

Possible extensions: Although palindromic cubes and palindromic fourth powers also exist $(111)^3 = 1367631$ and $(11)^4 = 14641$, they are harder to find. With the exception of $1^k = 1$, no palindromic k-th powers for $k > 4$ are yet (1982) known to your author. Some reader may find one.

Modem II Questions and Answers

Paul Schreiber
Radio Shack Research and Development

One of the handiest accessories a computer user can have is a modem. The Modem II, a Radio Shack top of the line modem, has many features that are not to be found in other modems. The most useful features are the auto answer/auto dial modes and two types of test modes. The Modem II has a custom microprocessor which makes things easy from a user's standpoint since the modem follows the micro's code, and the modem's operation is predictable. When programming the modem, some users have difficulty in getting it to "behave" properly. Perhaps the following questions and answers will give programmers a better understanding of the Modem II.

Q. WHAT IS THE PURPOSE OF THE FORCE DTR SWITCH?

A. The force DTR switch was installed to allow the Color Computer and other 3-wire RS-232 schemes to program the modem by use of the asterisk (*) command.

Q. WHAT ELSE DOES THE FORCE DTR SWITCH DO?

A. The DTR line is used as a master enable line and to place the modem in the programming mode. When the Force DTR switch is on

1. It forces CD (carrier detect). This is a Color Computer specification.
2. The modem's software driver cannot use CD to determine if another modem has answered.

See page 19 of the TRS-80 Modem II Operation Manual for further information on carrier detect.

3. It allows the Modem II to go into the programming mode only when it receives an * at the proper time.

The decision of when to use DTR has to be determined by the individual programmer based on his/her application.

Q. WHY WON'T THE MODEM II WORK WITH "XXX" BULLETIN BOARD SOFTWARE?

A. Software written to program a Brand X modem will not work with a Brand Y modem. Due to the manufacturing design differences in modems, Modem II software will not be not compatible with another manufacturer's modem. In fact, no two manufacturers modems program alike.

Q. THE MODEM II CAN'T BE PROGRAMMED THE WAY THAT I EXPECTED IT TO. WHAT SHOULD I DO?

A. Read the manual carefully, study the flowchart in the back of the manual, and write your program to conform to these instructions. If you do, the modem can be programmed without any difficulty.

The majority of the difficulties experienced with the Modem II stem from pre-conceived notions on how it should respond. Be sure that you are writing software to fit the modem and not trying to make the modem fit the software.

Q. HOW CAN I MAKE THE MODEM IGNORE INCOMING RINGS WHILE MY SOFTWARE IS DOING OTHER THINGS?

A. Following are two procedures to make the Modem II ignore incoming rings.

1. When the Modem is in LOCAL Test, the hardware resets the ring counter to zero so all incoming rings are ignored. To accomplish this under software control, send a QX (local test command) to start the test and then using software, drop DTR (the force DTR switch must already be off) to get out of local test.
2. If you must have force DTR on, then go into the programming mode. While in the programming mode, the modem will not process incoming rings. By sending an X, you will exit the programming mode, but remember that whatever is stored in the modem at the time the X is sent will be executed.

Q. I CAN'T PROGRAM THE MODEM WITH MY "XXX" TERMINAL SOFTWARE.

A. This problem usually occurs because of a lack of knowledge of the data format. The Modem II recognizes only upper case, 8 bit, no parity characters.

Q. WHEN USING FORCE DTR OFF, THE MODEM SOMETIMES SEEMS TO "HANG UP". WHY?

A. Usually when the modem appears dead in that it no longer responds to commands, it is because the computer is asserting or unasserting DTR for either the wrong amount of time or at the wrong time.

A good example of this may occur when the programmer uses DTR to disconnect the Modem II from the phone line after completing data transfer. When the DTR is dropped, the modem will promptly disconnect. Now, if DTR is kept low longer than 3 seconds (refer to the flowchart in the manual), the modem drops into the programming mode. If the unwary programmer fails to realize this and asserts DTR without first sending an X to exit the programming mode, the modem may appear to hang up. Be careful when using DTR to turn the modem on and off or you may accidently go into the programming mode.

If you wish to hang up the phone, it is better to send a space disconnect than it is to drop DTR. This eliminates the chance of accidentally going into the programming mode.

Q. HOW DO I GET THE SPACE DISCONNECT TO WORK?

A. Check to see what you are actually sending. The space disconnect does not refer to an ASCII space (20H) but a constant LOW output from the UART to the RS-232 drivers without start or stop bits. The internal microprocessor of the Modem II checks the status of the data every 75 ms during the 1.5 second and 3 second disconnect intervals. If during one of these times the data goes HIGH, then the timers are reset. Each computer/terminal's UART sends the space by different means, so check the technical specification of your system.

Personal Time Savers

VisiCalc command files can make your model building sessions quicker and more convenient.

Most people who turn to financial model building with the VisiCalc program say it saves them vast amounts of time when compared to the alternative methods of financial analysis with pencil, paper, and calculator. Yet you can save even more effort by spending a little time now thinking about frequently used formulas and often-repeated commands and converting them into VisiCalc command files.

How many times have you found during a model building session that you needed to insert ten rows — then had to key each insert in separately? How many times have you had to key in the same complex formula to a number of your models, because you thought there was no way of transferring a single formula from one worksheet to another? (Transferring values, by the way, from one sheet to another for consolidation is simple.)

You can go far towards eliminating this type of computer "grunt work" by using VisiCalc command files. When the VisiCalc program loads in a command file, it acts as though it were receiving commands typed at the keyboard. While command files can not do everything (for example, they cannot sequentially load a number of worksheets into one, or save worksheets for you), they can:

- perform multiple row and column inserts (at your cursor position)
- perform multiple row and column deletes
- format specified areas of your worksheet
- blank specified areas of your worksheet
- build a complex formula in a specified area of your worksheet

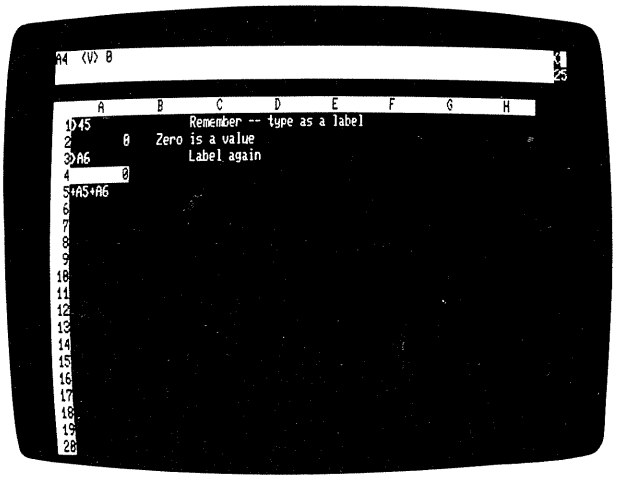
VISICALC®

BUILDING A COMMAND FILE

Creating a set of VisiCalc command files which suit your needs starts with the VisiCalc program and a blank, initialized data diskette. The data disk will hold your final library of VisiCalc command files, each one identified by a descriptive filename.

Let us start with a simple experiment which will show how well the VisiCalc program listens to your command files. We will type onto the blank VisiCalc sheet labels which resemble keyboard commands such as /F,)A5, etc. All these commands must be typed in as labels. If you wish to put a value into a formula (such as 0, for starters), it may be typed in as a value. Here is a simple addition formula which will show what we mean.

On a blank VisiCalc worksheet, bring your cursor to A1. Type the following:



Screen 1.

What we have done here is use the VisiCalc program to create text which looks like the kind of commands that would be typed into the computer keyboard. The row and column positions we are concerned with here are those appearing in the text. They will control where the command file appears when it is finally loaded in. We began writing at position A1 for convenience; the command file itself will begin the addition at A5, just as we commanded it to.

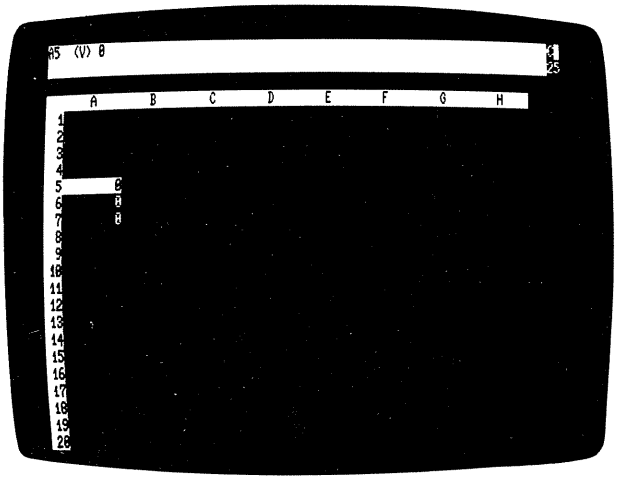
One cautionary note: While you may write your command file instructions from left to right on more than one row (one command to a cell position), you cannot write more than one column of commands vertically. This is because the VisiCalc program reads the commands from left to right until there are no more in a given row. Then it moves down to the next row and reads that from left to right. We have chosen to represent the command file as written down a column for clarity, but only one such column may be written.

Once our command file is written, we need to save it to diskette with the /PF command (Print to Filename), rather than the usual /SS (Storage Save). This alters the way the commands are stored on the diskette; the VisiCalc program will interpret them as commands (instead of labels).

Cautionary note: If you are planning to place a complex formula in a single cell, remember that /PF stores ONLY what appears on the VisiCalc screen. Make sure your column width reveals the entire formula, or the command file will not work when reloaded into the program.

In response to the program's request for a filename, name the file ADD, and give the LOWER RIGHT coordinate as A7.

Next, clear your worksheet with the /CY command. With /SL select the file ADD to be loaded. You should see a screen similar to the chart below.



Screen 2.

Your cursor should be positioned over A7, revealing the formula +A5+A6. Try different values in locations A5 and A6. The formula in A7 responds as though you had typed it in yourself . . . and in a sense, you have. That formula could have read @SUM(A5 . . . A6), or something even more complex.

Another note: the very nature of a formula such as +A5+A6 ties the answer to whatever is appearing in those two areas. However, if you have built your command file with no cursor direction commands (no ›A5, for example), the formula loads by default wherever your cursor happens to be. You can always /E (edit) the formula to refer to the proper cell positions; but be careful where your cursor is located when you do load a positionless formula onto your worksheet.

The same kind of command file creation can be used to build a library of often used calculations for incorporation into your models. If your often used formulas are of great complexity and do require a number of cells because of values referred to or sheer size, leave "formula area" at the top (or bottom, right, left, etc.) of all your worksheets. Then you will be able to write specific cursor positions (›A10) into your command file for all your formulas.

When loaded in, your library of formulas will always appear at the same open area and may then be moved or replicated to another area of your current worksheet.

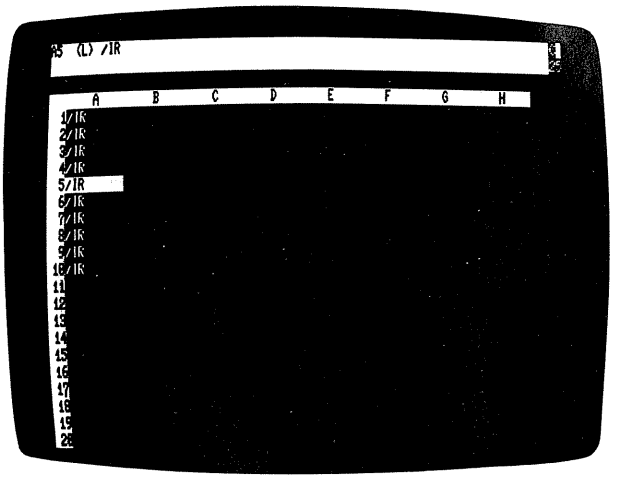
MORE BACKFLIPS AND SOMERSAULTS

Multiple row and column inserts can also be accomplished by VisiCalc command files. To free up ten rows, you would normally have to type the keystrokes / I R (in that sequence) ten times — 30 keystrokes — plus the few moments wait while your model moves down.

With a VisiCalc command file, you can reduce those keystrokes to / S L → until the correct filename shows, then ENTER. A command file for 10 row insertions is simple:

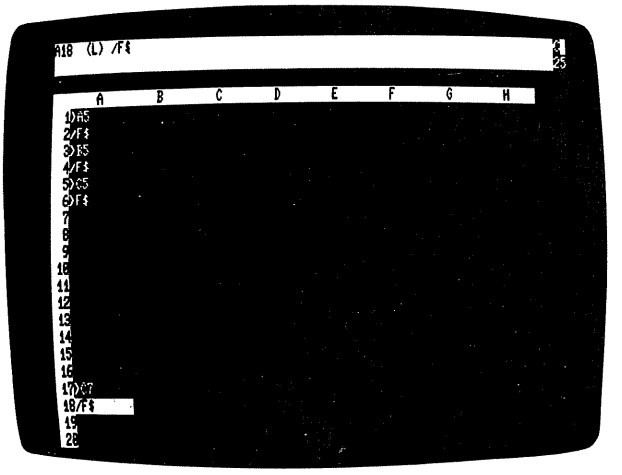
Remember that all these slash command entries must be typed as labels (preface each entry with a quotation mark " to trigger the label format).

Once again, save this INS10ROW file with the /PF command. As written, it will insert ten rows at whatever your current cursor position happens to be. With the addition of the label ›C27 at the beginning, for example, it can be made to insert ten rows only at location C27.



Screen 3.

If your models require that the two-decimal place dollar sign format (/F$) appear in an area bounded by A5, C5, A7 and C7, then you can write the following command file.



Screen 4.

At first this may appear laborious. Think though, that what you are typing here — only once — is the same sequence of keystrokes you would type in every model in which you needed them. Being able to format an area, rather than having to format globally and delete formats (/FD) or having to format cells individually over a large area is a time saver.

And that is what your TRS-80 and the VisiCalc program are all about, isn't it? Saving time.

If you are using VisiCalc command files to help with setting up and using VisiCalc models, we would like to hear about it, and may share your experience with other VisiCalc users here in the TRS-80 Microcomputer News.

Send a note to:

VisiCalc Page Editor
c/o Microcomputer News
P.O. Box 2910
Fort Worth, TX 76113-2910

Educational Programs

Customary-to-Metric

Todd Pruzan, Age 11
7920 Cindy Lane
Bethesda, MD 20034

Three months ago our family purchased a Model III. We have gotten great use out of it, and it is fun to work with. Now I have written a customary-to-metrics program. You can get a head start on the metric system with it. First it asks you several familiar words (miles, quarts, pounds, etc.). Type in the first letter—area is an exception to this—and it will ask you "MEASUREMENT?" Here you type the desired number, preferably a whole number from 1 to 100,000. Here you get an answer converted into metric. Lines 20 - 50 are skipped. You can use them up with REMs or PRINTs.

The sample run:

```
CUSTOMARY SYSTEM CONVERTED TO METRIC
BY TODD PRUZAN
LENGTH, AREA, MASS OR VOLUME? M __
OUNCES, POUNDS OR SHORT TONS? P __
MEASUREMENT? 17 __
7.65 KILOGRAMS.
ANOTHER CONVERSION? Y __
LENGTH, AREA, MASS OR VOLUME? A __
SQUARE INCHES, SQUARE FEET, SQUARE YARDS.
SQUARE MILES OR ACRES? SQUARE F __
MEASUREMENT? 9 __
.81 SQUARE METERS.
ANOTHER CONVERSION? N __
READY __
```

The program

```
10 CLS
20 PRINT"CUSTOMARY SYSTEM CONVERTED TO METRIC"
     : PRINT"BY TODD PRUZAN"
60 PRINT "LENGTH, AREA, MASS OR VOLUME";
70 INPUT M$
80 IF M$="L" THEN 120
90 IF M$="A" THEN 200
100 IF M$="M" THEN 260
110 IF M$="V" THEN 300
120 PRINT"WHAT DO YOU WANT CONVERTED?"
130 PRINT"INCHES, FEET, YARDS, OR MILES";
140 INPUT I$
150 PRINT"MEASUREMENT";
     : INPUT MT
160 IF I$="I" THEN PRINT MT*25;" MILLIMETERS."
     : GOTO 350
170 IF I$="F" THEN PRINT MT*30;" CENTIMETERS."
     : GOTO 350
180 IF I$="Y" THEN PRINT MT*.9;" METERS."
     : GOTO 350
190 IF I$="M" THEN PRINT MT*1.6;" KILOMETERS."
     : GOTO 350
200 PRINT "SQUARE INCHES, SQUARE FEET, SQUARE YARDS,
     SQUARE MILES OR ACRES"
     : INPUT SQ$ :PRINT"MEASUREMENT";
     : INPUT A
210 IF SQ$="SQUARE I" THEN PRINT A*6.5;" SQUARE
     CENTIMETERS."
     : GOTO 350
220 IF SQ$="SQUARE F" THEN PRINT A*.09;" SQUARE
     METERS."
     : GOTO 350
230 IF SQ$="SQUARE Y" THEN PRINT A*.8;" SQUARE
     METERS."
     : GOTO 350
240 IF SQ$="SQUARE M " THEN PRINT A*2.6;" SQUARE
     KILOMETERS."
     : GOTO 350
250 IF SQ$="A" THEN PRINT A*.4;" SQUARE HECTOMETERS."
     : GOTO 350
260 PRINT "OUNCES, POUNDS, OR SHORT TONS ";
     : INPUT O$
265 PRINT"MEASUREMENT";
     : INPUT B
270 IF O$="O" THEN PRINT B*28;" GRAMS."
     : GOTO 350
280 IF O$="P" THEN PRINT B*.45;" KILOGRAMS."
     : GOTO 350
290 IF O$="S" THEN PRINT B*.9;" ,MEGAGRAMS (OR METRIC
     TONS)."
     : GOTO 350
300 PRINT"OUNCES, PINTS, QUARTS, OR GALLONS";
     : INPUT P$
     : PRINT "MEASUREMENT";
     : INPUT C
310 IF P$="O" THEN PRINT C*30;" MILLILITERS."
     : GOTO 350
320 IF P$="P" THEN PRINT C*.47;" LITERS."
     : GOTO 350
330 IF P="Q" THEN PRINT C*.45;"LITERS."
     : GOTO 350
340 IF P$="G" THEN PRINT C*3.8;"LITERS."
350 PRINT "ANOTHER CONVERSION";
360 INPUT X$
370 IF X$="Y" THEN 60
380 END
```

Physical Science Program

Allan Freeman, Age 14
100 5th Place
Pleasant Grove, Alabama 35127

I own a Model III and a Line Printer VII. I wrote this program to show how helpful a computer can be when your studying for a test.

```
2 CLS
5 REM
     : This program was written for the Model III
10 REM December 5, 1981 (No Rights Reserved)
15 REM Written by Allan Freeman
20 POKE 16916,4
22 POKE 16412,1
25 PRINT TAB(20);"Physical Science (True or False)"
26 PRINT TAB(20);"Written by: Allan Freeman"
27 CLEAR 100
28 PRINT STRING$(64,191)
30 INPUT "1. A metric ruler, graduated cylinder, and
     microscope are tools used in science. (T/F)";A$
32 IF A$="T" THEN GOSUB 200
34 IF A$="F" THEN GOSUB 400
40 INPUT"2. A hypothesis is always true. (T/F)";A$
42 IF A$="T" THEN GOSUB 400
44 IF A$="F" THEN GOSUB 200
50 INPUT "3. The metric system of measurement is a
     decimal system. (T/F)";A$
52 IF A$="T" THEN GOSUB 200
```

```
54 IF A$="F" THEN GOSUB 400
60 INPUT"4. A scientific law can never be disproved.
     (T/F)"; A$
62 IF A$="T" THEN GOSUB 400
64 IF A$="F" THEN GOSUB 200
70 INPUT "5. Metric units have international use in'
     science. (T/F)";A$
72 IF A$="T" THEN GOTO 218
74 IF A$="F" THEN GOTO 418
200 CLS:PRINT TAB(20);"C O R R E C T ! ! !"
210 FOR X=1 TO 1000
215 NEXT X
     : CLS
     : RETURN
218 CLS
219 PRINT "C O R R E C T !
     C O R R E C T !
     C O R R E C T !
     C O R R E C T !
220 PRINT TAB(15);"YOU'RE ALMOST AS SMART AS I AM."'
221 GOTO 600
400 CLS
     : PRINT TAB(20);"W R O N G ! ! !"
410 FOR X=1 TO 1000
415 NEXT X
     : CLS
     : RETURN
418 CLS
419 PRINT TAB(5);"YOU MISSED THAT ONE."
     : INPUT"DO YOU WISH TO TRY AGAIN?";D$
420 IF D$="Y" GOTO 30
421 IF D$="N" GOTO 600
600 POKE 16916,0
     : POKE 16412,0
     : PRINT TAB(20); "ADIOS"
     : PRINT TAB(20); "END OF PROGRAM"
     : STOP
```

Directory Algorithm

Thomas J. Kelanic
935 Lilly Lane
Pittsburgh, PA 15221

I became interested in multiple-page directories while writing a computer program to generate daily absence bulletins for my school. Under the old system, a secretary collected attendance rosters from all homerooms, then typed a list of absent students. The list was divided into separate sublists, one for each homeroom. A teacher wishing to check-up on a student first had to know, or find elsewhere, the student's homeroom number. After finding the sublist for the given homeroom, the teacher then looked for the student's last name, alphabetically.

I wanted to improve on the old system by generating a daily directory that looked like a telephone directory. All the inquiring teacher would have to know would be the last name of the student, since there would be only one alphabetized list of names. Following each name would be whatever other information was required, such as homeroom number and cumulative absences. To avoid the possible future necessity of having to go back and re-write my program to take into account changed requirements, I decided that the number of pages, columns, lines per page, and page dimensions should all be variable.

The following program does not produce daily absence bulletins. It will produce any multiple-page directory listing required from a string array in RAM. I believe my program is a general algorithm for producing directories.

In lines 100 - 130, the user's string array is simulated in RAM. The 676 strings "AA—" through "ZZ—" are produced in approximately one minute and forty seconds. These lines must be omitted if the user has their own A$ array in RAM.

In lines 170 - 230, not counting 220, the given environmental constants are set. These lines could easily be replaced by LETs. Lines 230 - 290 are the program variables used to specify the directory required, and are the parameters used in the examples which follow. In a real application, N, in line 230, would be constant. However, in this demonstration program N is variable, since the user may not wish to see all 676 strings.

No error-checking is provided in my program. If the user asks for something physically impossible, the program will "bomb". In this case, the user may issue a "GOTO 230" command, and should reconsider his inputs.

To get an idea of the flexibility of the program, the user may try the following two examples. Assuming the constants input were "actual max." = 5, "highest (fixed)" = 63, "lowest (fixed)" = 0, the user is ready.

Example #1: Use 100 strings, 63 highest, 34 lowest, 5 width, 6 columns, and 15 lines per page.

Example #2: Use of 61 strings, 63 highest, 0 lowest, 30 width, 2 columns, and 10 lines per page.

To guarantee at least one space between columns, use a field-width one more than the actual maximum string length. For the example strings of length 5 a field-width of 6 will do.

If you have a line-printer, you must use the correct values for your printer in lines 180 - 210, and change the two PRINTs in lines 600 and 610 to LPRINTs. Also, you may wish to insert some LPRINTs between lines 630 and 640.

```
10 '
20 '            DIRECTORY ALGORITHM
30 '
40 '            THOMAS J. KELANIC
50 '            935 LILLY LANE
60 '            PITTSBURGH, PA. 15221
70 '
80 '            SIMULATE USER ARRAY A$
90 '
100 CLEAR 6000
     : T6=26

     : DIM A$(T6*T6)
110 FOR I=1 TO T6
     : FOR J=1 TO T6
120 A$((I-1) * T6+J)=CHR$(I+64) + CHR$(J+64)+"---"
130 NEXT J
     : NEXT I
140 '
150 '             GET PARAMETERS
160 '
170 INPUT "ACTUAL MAX. STRING LENGTH"; W2
180 PRINT "HIGHEST PRINT POSITION ";
190 INPUT "(FIXED) FOR OUTPUT DEVICE"; W9
200 PRINT "LOWEST PRINT POSITION ";
210 INPUT "(FIXED) FOR OUTPUT DEVICE"; W0
220 DIM B(INT((W9-W0+1)/W2)+1)
230 INPUT "NUMBER OF STRINGS TO BE USED"; N
240 INPUT "HIGHEST PRINT POSITION TO BE USED"; W8
250 INPUT "LOWEST PRINT POSITION TO BE USED"; W1
260 INPUT "COLUMN FIELD-WIDTH TO BE USED"; W
270 PRINT "SELECT 1 TO"; INT((W8-W1+1)/W);
280 INPUT "COLUMNS"; C
290 INPUT "NUMBER OF LINES PER PAGE"; L
300 '
310 '            FOR EACH DIRECTORY
320 '
330 K=L
     : R=0
     : P=INT((N-1)/(C*L))+1
340 CLS
     : PRINT P;"PAGES WILL BE PRODUCED"
350 PRINT "PRESS (ENTER) FOR NEXT PAGE"
360 FOR I=1 TO 600
     : NEXT I
370 '
380 '            FOR EACH PAGE
390 '
400 FOR H=1 TO N STEP C*L
```

```
41Ø CLS
     : B(1)=H
     : P2=INT((B(1)-1)/(C*L))+1
42Ø PRINT "PAGE NUMBER"; P2
43Ø FORI=1 TO 4ØØ
     : NEXT I
     : CLS
44Ø IF P2=P THEN D=N+1-B(1)
     : K=INT(D/C)
     : R=D-K*C
45Ø '
46Ø '            SET BREAKPOINTS
47Ø '
48Ø FORI=1 TO C
     : B(I+1)=B(I)+K
49Ø IFI <= R THEN B(I+1)=B(I+1)+1
5ØØ NEXT I
51Ø '
52Ø '            OUTPUT PAGE
53Ø '
54Ø FOR I=Ø TO K
     : FOR J=1 TO C
55Ø IF I+B(J)=B(J+1) THEN 63Ø
56Ø T=INT(((W8-W1+1-C*W)/(C+1))*J+(J-1)*W+W1))
57Ø '
58Ø '            OUTPUT STRING
59Ø '
6ØØ PRINT TAB(T);A$(I+B(J));
61Ø NEXT J
     : IF T+W-1 < W9 THEN PRINT
62Ø NEXT I
63Ø X$=INKEY$
     : IF X$=""THEN 63Ø
64Ø NEXT H
     : PRINT
     : GOTO 23Ø
```

String Formatting and the Window Game

Doug Fuge
2427 Hoohoihoi Street
Pearl City, HI 97682

Having just received my February issue of the newsletter and enjoying some of the programs, I thought I would comment on one article and send another of the programs I wrote for my youngest daughter. Both she and I got a big kick out of seeing our names in print in the December newsletter. Furthermore, it was rewarding to receive two inquiries from readers who had tried the Character Recognition program and were having minor difficulties getting it to run properly (Note: the rendition in the newsletter was accurate - cockpit errors were the problems).

The article More PRINT USING was most interesting to me as I am working on several programs where formatting is the key. While working with the concept advanced by Johnny Bond, and the various examples of the use of the TAB() function, I found I could utilize the STRING$ (,) function to insert spacing into the format string. Attached is Program #2 of Johnny Bond's article with line 20 changed to use the STRING$ (10,32) to indent the output on both the Video and my Line Printer IV. I am sure other readers really appreciate the power of the concept advanced in that article.

```
1Ø CLS
2Ø S$=STRING$(1Ø,32)+"NAME %          %
     "+STRING$(1Ø,32)+"BIRTHDATE : ##/##/##
     "+STRING$(1Ø,32)+"TELEPHONE : (###) ### ####"
3Ø INPUT"NAME";N$
4Ø INPUT"YEAR OF BIRTH (2 DIGITS)";YB
5Ø INPUT"MONTH OF BIRTH (NUMBER)";MB
6Ø INPUT"DAY OF BIRTH";DB
7Ø INPUT"AREA CODE";AC
8Ø INPUT"PHONE NUMBER (FORMAT#######)";PN
9Ø T1=INT(PN/1ØØØØ)
1ØØ T2=PN-(T1*1ØØØØ)
11Ø PRINT
12Ø PRINTUSINGS$;N$,MB,DB,YB,AC,T1,T2
125 LPRINTUSINGS$;N$,MB,DB,YB,AC,T1,T2
13Ø PRINT
14Ø GOTO 3Ø
```

When Line 20 is entered, use the down arrow to move to the second line after entering the second percent sign. Use the down arrow again after the sixth # sign following "BIRTHDATE : " to move to the next line.

Although I have not tried the concept of the Instant Recall program by Dwight Dager in the February issue, it brought to mind a game my current five-year-old has enjoyed for almost two years. It is like Concentration and she calls it her "Window Game" because of the screen format. It did not take many times for her to memorize the words and I feel the word recognition has assisted her in learning to read. I wrote the program on my Model I Level II TRS-80.

Bracken's Window Game

```
9Ø CLS
     : PRINT "WINDOW GAME FOR BRACKEN"
1ØØ DIM N$(16), W$(16), W(16)
11Ø FOR N=1 TO 16
     : READ N$(N)
     : W$(N)=""
     : NEXT N
12Ø FOR N=1 TO 8
121 READ W$(Ø)
13Ø N1=RND(16)
     : IF W$(N1) <> "" THEN 13Ø ELSE W$(N1)=W$(Ø)
14Ø N2=RND(16)
     : IF W$(N2) <> "" THEN 14Ø ELSE W$(N2)=W$(Ø)
16Ø NEXT N
17Ø GOSUB 1ØØØ
18Ø GOSUB 2ØØØ
2ØØ T=1
     : C=Ø
21Ø GOSUB 3ØØØ
4ØØ IF W$(N1)=W$(N2) THEN 5ØØ
41Ø FOR Z=Ø TO 1ØØØ
     : NEXT Z
42Ø GOSUB 2ØØØ
43Ø T=T + 1
     : PRINT@ 97Ø, CHR$(3Ø);
45Ø GOTO 21Ø
5ØØ REM
52Ø PRINT@ 97Ø, CHR$(3Ø);
     : PRINT@ 97Ø, "MATCH";
525 FOR Z=Ø TO 6ØØ
     : NEXT Z
53Ø W$(N1)=""
     : W$(N2)=W$(N1)
     : N$(N1)="   "
     : N$(N2)=N$(N1)
535 FOR Z =1 TO 16
     : IF W$(Z) <> "" THEN 54Ø ELSE NEXT Z
537 PRINT@ 97Ø, "GOOD JOB";
538 E$=INKEY$
     : IF E$=<> "" THEN RUN ELSE 538
54Ø C=C+1
     : T=T+1
     : GOSUB 2ØØØ
55Ø GOTO 21Ø
1ØØØ CLS
     : PRINT CHR$(23)
1Ø1Ø FOR Y1=Ø TO 11
     : FOR X1=Ø TO 46 STEP2
1Ø2Ø X=15434 + 64 * Y1 + X1
     : POKE X,191
1Ø3Ø NEXT X1, Y1
1Ø4Ø RETURN
2ØØØ N=1
     : FOR Y1=Ø TO 3
     : FOR X1=Ø TO 3
2Ø1Ø X=142 + 192 * Y1 + 1Ø * X1
2Ø2Ø PRINT@X, N$(N);
     : W(N)=X
     : N=N+1
```

```
2030 NEXT X1, Y1
2040 RETURN
3000 PRINT@ 906,T;" TRIES  ";C" CORRECT";
3010 PRINT@ 970,"WINDOW?";
     : O=1
     : GOTO 3020
3015 PRINT@ 990," AND
     : ";
3020 G$=INKEY$
     : IFG$="" THEN 3020 ELSE PRINT G$;
3030 H$=INKEY$
     : IF H$="" THEN 3030 ELSE PRINT H$;
3035 I$=" "+G$+H$
3040 FOR N=1 TO 16
     : IF N$(N)=I$ THEN 3060 ELSE NEXT N
3050 IF O=1 THEN PRINT@ 970, CHR$(30);
     : GOTO 3010
3055 PRINT@ 990, CHR$(30);
     : GOTO 3015
3060 IF O <> 1 THEN 3080
3070 O=O+1
     : N1=VAL(G$+H$)
     : Q=N1
     : GOSUB 4000
     : N1=Q
     : GOTO 3015
3080 N2=VAL(G$+H$)
     : Q=N2
     : GOSUB 4000
     : N2=Q
     : IFN2=N1 THEN 3055
3090 RETURN
4000 IF Q<20 THEN Q=Q-10
     : GOTO 4070
4010 IF Q<30 THEN Q=Q-20
     : GOTO 4060
4020 IF Q<40 THEN Q=Q-30
     : GOTO 4050
4030 Q=Q-40
4040 Q=Q+12
     : GOTO 4070
4050 Q=Q+8
     : GOTO 4070
4060 Q=Q+4
4070 PRINT@ W(Q), W$(Q);
4080 RETURN
9000 DATA" 11"," 12"," 13"," 14"," 21"," 22"," 23","
     24"," 31"," 32"," 33"," 34"," 41"," 42"," 43","
     44"
9010 DATA "CAT", "DOG", "HAT", "DAD", "MAT", "RAT",
     "MAD", "LOG"
```

Keep the good news coming. I look forward to each issue.

Crush—A Mathematical Game

Terry Myerson, Age 9
233 Pine Cone Trail
Ormond Beach, FL 32074

```
0 '********************
1 '** TERRY MYERSON ***
2 '*** GRADE 4,AGE 9 **
3 '** ORMOND BEACH  ***
4 '*** FLORIDA 32074 **
5 '********************
8 CLS
10 C=76
     : D=43
     : P=0
15 GOSUB 8000
16 GOSUB 4000
17 GOSUB 30000
20 FOR X=40 TO 80
     : SET(X,18)
     : NEXT X
30 FOR Y=18 TO 47
     : SET(40,Y)
     : SET(39,Y)
     : NEXT Y
40 FOR Y=18 TO 47
     : SET(80,Y)
     : SET(79,Y)
     : NEXT Y
50 FOR X=1 TO 127
     : SET(X,47)
     : NEXT X
55 IF HH=0 THEN GOTO 70
60 FOR X=1 TO 127
     : SET(X,15)
     : NEXT X
65 IF HH=1 THEN GOTO 100
70 PRINT @994-14,STRING$(20,191)
75 HH=1
     : GOTO 60
100 PRINT @128+64,"     (36 spaces)
    ":GOSUB 1000
102 PRINT @24,"*** CRUSH ***"
110 P=P+1
     : PRINT @128+64,"PROBLEM #";P;"   -  ";
    A;"";UI$;"";B;"=";
120 INPUT G
125 IF UI$="+" THEN IF G<>A+B THEN GOSUB 2000
126 IF UI$="-" THEN IF G<>A-B THEN GOSUB 2000
127 IF UI$="X" THEN IF G<>A*B THEN GOSUB 2000
130 GOTO 100
1000 A=RND(TT)
     : B=RND(TT)
1010 IF UI$="-" AND A<B THEN GOTO 1000
1020 RETURN
1520 RETURN
2000 IF C=67 THEN GOTO 50000
2005 FOR Y=16 TO 44
2010 SET(C,Y)
     : SET(D,Y)
2020 NEXT Y
2025 C=C-3
     : D=D+3
2030 RETURN
4000 CLS
     : PRINT @64*4,"WHAT TYPE OF PROBLEMS WOULD YOU
    LIKE"
4010 PRINT @64*5,"- SUBTRACTION, X MULTIPLICATION, +
    ADDITION"
4020 INPUT"PLEASE TYPE X,+,-";UI$
4025 CLS
4030 RETURN
8000 PRINT @64*6,"UP TO WHAT # TABLES WOULD YOU LIKE
    ";
8010 INPUT TT
8020 IF TT >= 30000 PRINT "TOO HIGH"
     : IF TT >= 30000 THEN GOTO 8010
8030 CLS
8040 RETURN
30000 PRINT@473 + 64, STRING$(4,128) + CHR$(159) +
    CHR$(155) + CHR$(149)
30010 PRINT @473 + 128, CHR$(144) + STRING$(3,128) +
    CHR$(131) + CHR$(151) + CHR$(129) +
    STRING$(2,128) + CHR$(160)
30020 PRINT@64 * 3 + 473, STRING$(2,143) + CHR$(189)
    + CHR$(190) + STRING$(2,191) + CHR$(189) +
    CHR$(190) + STRING$(2,143)
30030 PRINT@64 * 4 + 473, STRING$(3,128) + CHR$(171)
    + STRING$(2,191) + CHR$(151)
30040 PRINT@64 * 5 + 473, STRING$(2,128) + CHR$(160)
    + CHR$(186) + STRING$(2,191) + CHR$(181) +
    CHR$(144)
30050 PRINT@64 * 6 + 473, STRING$(2,128) + CHR$(170)
    + CHR$(191) + CHR$(128) +
    CHR$(128)CHR$(191)CHR$(149)
30070 PRINT@64 * 7 + 473, STRING$(2,128) + CHR$(138)
    + CHR$(159)CHR$(128)CHR$(128)CHR$(175)CHR$(133)
30080 PRINT@64 * 8 + 473, STRING$(2,128) + CHR$(130)
    + CHR$(131) + CHR$(128) + CHR$(128) + CHR$(131)
    + CHR$(129)
40000 RETURN
50000 CLS
50010 PRINT"YOUR LIFE WAS ";P;" PROBLEMS LONG"
50020 IF P > 20 THEN PRINT "I CAN TELL YOU KNOW YOUR
    FACTS YOU ONLY MISSED 3 OUT OF ";P
50030 IF P < 20 THEN PRINT "YOU NEED TO LEARN YOUR
    FACTS"
50040 PRINT
     : PRINT
50050 INPUT "DO YOU WANT TO PLAY AGAIN";GA$
50060 IF LEFT$(GA$,1)="Y" THEN RUN
50070 CLS
     : CMD"S"
```

Prime Number Generator #1

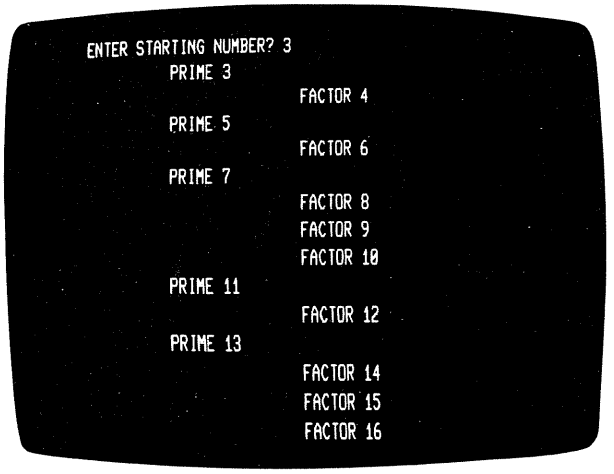
Gary M. Greer
31923 Angela Drive
Goshen, IN 46526

I wrote the following program on my TRS-80 Model I computer as a prime and factor number generator.

You are asking why would anyone be interested in primes? Well, I felt this would be an interesting task for a beginner such as myself.

Enough said. Well, okay . . . go to it Model I, spit out those primes!

```
1Ø 'PRIME NUMBER GENERATOR
2Ø 'BY GARY GREER...GOSHEN, IN.
3Ø CLS
4Ø INPUT "ENTER STARTING NUMBER"; N
5Ø FOR A=2 TO N
6Ø IF N<2 THEN N=N+1
     : GOTO 5Ø
7Ø Q=N/A
8Ø IF Q=1 THEN 11Ø
9Ø IF INT(Q)*A=N THEN 12Ø
1ØØ NEXT
11Ø PRINT TAB(1Ø) "PRIME" N
     : N=N+1
     : GOTO 5Ø
12Ø PRINT TAB(25) "FACTOR" N
     : N=N+1
     : GOTO 5Ø
```



Prime Number Generator #2

Russ Euliano
201 Sweetwater Cove Blvd. N.
Longwood, FL 32750

This program is something I have been playing with for a while and I seem to have found a very fast way to find or generate prime numbers. No, this will not help you with your payroll but I find it fascinating that the TRS-80 can find all the primes between 1 and 1,000 in only 23 seconds. This task could take days or weeks using a pencil and paper.

This program is short but fairly involved so I will do my best to explain the logic of the program. This is the program. I call it PRIME/BAS.

```
1Ø  DEFINT A-Z
2Ø  INPUT "ENTER THE NUMBER TO STOP SEARCHING FOR
     PRIMES"; Q
3Ø  IF MEM<Q*2+2ØØ THEN PRINT "TOO HIGH.  TRY AGAIN"
     : GOTO 2Ø
4Ø  DIM A(Q)
5Ø  FOR X=2 TO Q/2
6Ø  IF A(X)=1 THEN GOTO 11Ø
7Ø  PRINT USING "##,###  "; X;
8Ø  FOR Y=2*X TO Q STEP X
9Ø  A(Y)=1
1ØØ NEXT Y
11Ø NEXT X
12Ø FOR X=Q/2+1 TO Q
13Ø IF A(X)=Ø THEN PRINT USING "##,###  "; X;
14Ø NEXT X
```

The basic principal in this program is to start with 2 and find all the numbers with 2 as a factor and set that element of the array equal to 1 which means it is not a prime. It then moves on to three and does the same thing. Since 4 is a factor of 2 it would be a waste of time to find all the numbers with 4 as a factor since all of them would have already been ruled out by the number 2.

The FOR X-NEXT X loop contains the number in which the program searches for all numbers which have it as a factor and only goes half way to the ending number since no number more than half way could be a factor of any number between 1 and the ending number.

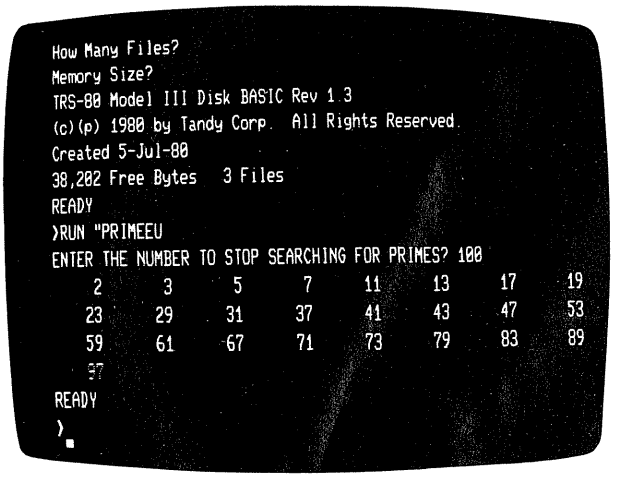
Line 60 checks to see if the number X has any other number as a factor, and if it does, it skips to LINE 110 which gets a new number.

If the number doesn't have any factors, then it is a prime and is printed out using an 8 space format so the numbers will be printed in nice, neat columns.

The Y loop is next and goes from two times the number X (the next number with X as a factor) to the ending number and counts by the number X to find all the numbers with X as a factor. When the X loop is done all the the primes between 1 and the ending numbers have been found and all the primes between 1 and half way to the ending number have been printed so the only thing left is to print out the rest.

I then simply set up a FOR NEXT loop between 1 more than half way and the end which prints all the elements that are equal to zero (have no factors).

Actually this program finds all the numbers that AREN'T primes and prints all the others.



Arithmetic Practice

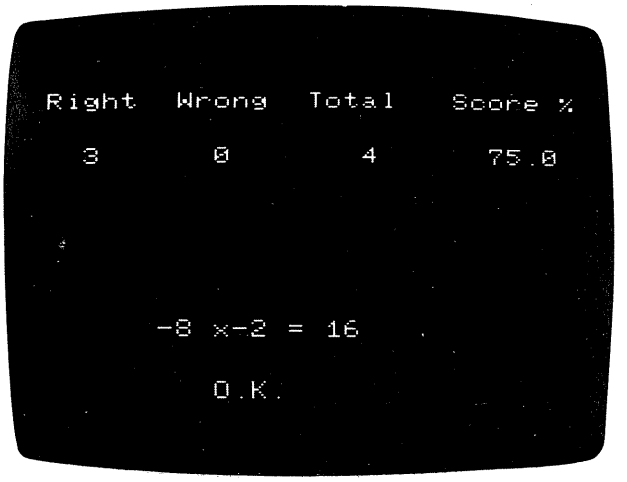
Peter Fried
17 Hearthstone Circle
Scarsdale, NY 10583

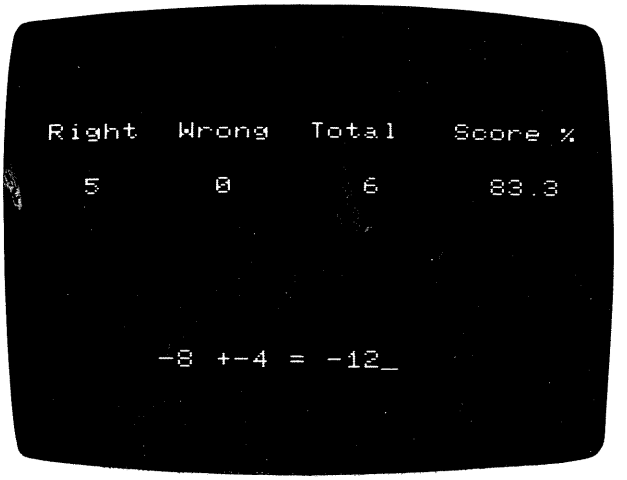
I designed the following little Model III Disk BASIC program for my son Kenny. It allows him to work Addition, Subtraction, Multiplication and Division problems at random using integers between -10 and +10.

The program also demonstrates the use of the Video scroll control, cursor character control, as well as the ⟨**ENTER**⟩ key control (PEEK 14400). The score is constantly displayed on top of the screen.

Note: the program will run in a non-disk Model III by changing the LINE INPUT statement in line 1020 to INPUT.

```
2 REM PROGRAM NAME : PRACTICE
5 CLS
     : PRINT @ 261, "This program generates Addition,
     Subtraction,"
     : PRINT "    Multiplication and Division
     problems at Random."
     : PRINT
     : PRINT TAB(6) "Hit "; CHR$(34); "ENTER";
     CHR$(34); " to Start. ";
6 PRINT TAB(16) "Enter "; CHR$(34); "9999"; CHR$(34);
     " to quit."
     : PRINT @ 963, "Program by Peter Fried .....
     February 1Ø, 1982";
1Ø IF PEEK(14400)-1 THEN 1Ø ELSE CLS
     : PRINT CHR$(23)
     : RANDOM
15 POKE 16419, 95
     : POKE 16916, 3
     : PRINT @ 6, "Right  Wrong  Total   Score %"
2Ø D=Ø
     : C=RND(4)
22 A=RND(2Ø)-1Ø
     : B=RND(2Ø)-1Ø
     : PRINT @ 512,;
25 IF A=Ø THEN 2Ø ELSE IF B=Ø THEN 2Ø ELSE IF
     A/B<>INT(A/B) THEN 22
3Ø T=T+1
     : ON C GOTO 1ØØ, 2ØØ, 3ØØ, 4ØØ
1ØØ REM ..... ADDITION .....
11Ø GOSUB 1Ø1Ø
     : PRINT @ 53Ø, A; "+"; B; "= ";
     : GOSUB 1Ø2Ø
     : IF X=A+B THEN 1ØØØ
125 D=D+1
     : IF D=1 THEN W=W+1
13Ø GOTO 11Ø
2ØØ REM ..... SUBTRACTION .....
21Ø GOSUB 1Ø1Ø
     : PRINT @ 53Ø, A; "-"; B; "= ";
     : GOSUB 1Ø2Ø
     : IF X=A-B THEN 1ØØØ
225 D=D+1
     : IF D=1 THEN W=W+1
23Ø GOTO 21Ø
3ØØ REM ..... MULTIPLICATION .....
31Ø GOSUB 1Ø1Ø
     : PRINT @ 53Ø, A; "x"; B; "= ";
     : GOSUB 1Ø2Ø
     : IF X=A*B THEN 1ØØØ
325 D=D+1
     : IF D=1 THEN W=W+1
33Ø GOTO 31Ø
4ØØ REM ..... DIVISION .....
41Ø GOSUB 1Ø1Ø
     : PRINT @ 53Ø, A; "/"; B; "= ";
     : GOSUB 1Ø2Ø
     : IF X=A/B THEN 1ØØØ
425 D=D+1
     : IF D=1 THEN W=W+1
43Ø GOTO 41Ø
1ØØØ PRINT
     : PRINT TAB(12) "O.K."
     : IF D=Ø THEN R=R+1
1ØØ5 FOR I=1 TO 2ØØ
     : NEXT
     : PRINT @ 136, USING "##      ##      ###
     ###.# "; R, W, T, R*1ØØ/T;
     : PRINT @ 576,;
     : PRINT
     : GOTO 2Ø
1Ø1Ø PRINT @ 53Ø, CHR$(22Ø);
     : RETURN
1Ø2Ø LINE INPUT X$
     : IF X$="9999" THEN POKE 16916, Ø
     : CLS
     : END
1Ø25 X=VAL(X$)
     : RETURN
```





Head Injury

Southern New Jersey Head Injury Support Group
1316 Beaverbrook Drive
Cherry Hill, NJ 08034

The above group is seeking information on computer programs relating to head injuries or listings thereof. If any individual or association has such information please contact the Southern New Jersey Head Injury Support Group at the address shown.

Model I/III Bugs, Errors, and Fixes

NOTE TO USERS:

The following program changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that the changes should be made, but you do not feel qualified to make the changes yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Services in Fort Worth for assistance.

CHANGES TO BASIC PROGRAMS

There are general procedures that need to be followed when any corrections are made.

1. Make a backup of the tape or disk that contains the program to be corrected. Changes should be made on the backup copy.
2. Load the program to be changed by typing CLOAD "filename" (tape) or LOAD"filename" (disk) where filename is the name of the program to be modified.
3. Make the line changes indicated in the fix. For existing line numbers, edit or retype the line to match the one in the fix. Enter new lines.
4. Save the corrected program (the one now in memory). Type CSAVE"filename"⟨ENTER⟩ (tape) or SAVE "filename" ⟨ENTER⟩ (disk) where filename is the name of the program that has been modified.
5. Now make a backup of the corrected tape or diskette.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHES to the backup copy of the diskette.
2. Apply PATCHES according to the information given in your TRSDOS manual.

CASSETTE PORTFOLIO (26-1506)

The sell calculations in the option program are reversed. To correct the problem CLOAD the Option tape and make the following changes. (Model III Version 1.0)

Line: `340 FORI=1TONO:GOSUB800:NEXT`

Add these lines:

```
800 CV#=O5#(I)*O(I,2)*100:REM CURRENT VALUE
810 OV#=O(I,0)*O(I,2)*100:REM OPTION VALUE
820 OC#=O4#(I):REM OPTION COST
830 IFO3$(I)="B"THENGL#=CV#-OV#-OC#ELSEGL#=
    OV#+OC#-CV#
840 O6#(I)=GL#:RETURN
```

CSAVE to save the changes.

STOCKPAK (26-1507)

Below are two sets of program changes to correct two separate problems with Stockpak.

1—Erratic performance, including syntax errors, reboot, etc., is exhibited when printing reports. Make the following corrections to the program PTREPORT. (Model III Version 3.0)

Line: `350 IFPEEK(16425)>50THENLPRINTCHR$(12);:POKE16425,1:
Q%=Q%+1ELSERETURN`

ADD line 7:

```
7 POKE 16412,255
```

SAVE"PTREPORT" to store the program changes.

Additional corrections need to be made to the program REPRUN/BAS.

Add the following lines.

Line: `5 POKE16412,255`

Line: `1503 POKE16425,1`

SAVE"REPRUN/BAS" to store the program changes.

2—The Eligibility for Long Term Gains Report is inaccurate when stocks with no commission are included. (Model III Version 3.0)

Make the following change to the program PTREPORT.

Line: `2520 FORI=K8TOK7:FL=0:CM#=0#:FORI1=1TO3:FORI2=0TO9`

SAVE"PTREPORT" to store the program changes.

ACCOUNTS PAYABLE (26-1554)

Following are four sets of changes to correct separate problems with Accounts Payable.

1—Whether the end of period is performed or not, the number of invoices selected and posted is inaccurate after loading the end of period processing program. (Model I Version 3.0)

Make the following correction to the program PROCESS.

Add line 124:

```
124 LSETU1$(1)=MKI$(I):LSETU1$(5)=MKI$(IS)
```

SAVE"PROCESS" to store the program change.

2—During End-of-Period Processing Error Code 5 in line 45 can occur. (Models I/III Version 3.0 and Prior)

Make the following changes to the program PROCESS.

VERSIONS PRIOR TO 3.0

Line: `350 W#=ABS(N#)*100+.1:X=W#/D1#:W#=W#-X*D1#
:V$=CHR$(X-(N#<0)*128):X=W#/D2#
:W#=W#-X*D2#:V$=V$+CHR$(X):X=W#/D3#
:W#=W#-X*D3#:V$=V$+CHR$(X)+CHR$(W#)
:RETURN`

Line: `2530 PRINT:PRINTCHR$(30);" UNEXPECTED ERROR CODE
";ERR/2+1; "IN LINE";ERL:GOSUB490:END`

SAVE"PROCESS" to store the program changes.

Make the following changes to the program CHECKS.

Line: `350 W#=ABS(N#)*100+.1:X=W#/D1#:W#=W#-X*D1#
:V$=CHR$(X-(N#<0)*128):X=W#/D2#:W#=W#-X*D2#
:V$=V$+CHR$(X):X=W#/D3#:W#=W#-X*D3#
:V$=V$+CHR$(X)+CHR$(W#):RETURN`

Line: `2530 PRINT:PRINTCHR$(30);" UNEXPECTED ERROR CODE ";
ERR/2+1; "IN LINE";ERL:GOSUB490:END`

SAVE"CHECKS" to store the program changes.

Make the following additional changes to the program POST.

Line: `350 W#=ABS(N#)*100+.1:X=W#/D1#:W#=W#-X*D1#
:V$=CHR$(X-(N#<0)*128):X=W#/D2#:W#=W#-X*D2#
:V$=V$+CHR$(X):X=W#/D3#:W#=W#-X*D3#
:V$=V$+CHR$(X)+CHR$(W#):RETURN`

Line: `2530 PRINT:PRINTCHR$(30);" UNEXPECTED ERROR CODE ";
ERR/2+1; "IN LINE";ERL:GOSUB490:END`

SAVE"POST" to store the program changes.
Make the following changes to the program INVOICES.

Line:
```
35Ø N#=VAL(IN$):W#=ABS(N#)*1ØØ+.1:X=W#/D1#
:W#=W#-X*D1#:V$=CHR$(X-(N#<Ø)*128):X=W#/D2#
:W#=W#-X*D2#:V$=V$+CHR$(X):X=W#/D3#
:W#=W#-X*D3#:V$=V$+CHR$(X)+CHR$(W#):RETURN
```

Line:
```
253Ø PRINT:PRINTCHR$(3Ø);" UNEXPECTED ERROR CODE ";
ERR/2+1; "IN LINE";ERL:GOSUB49Ø:END
```

SAVE"INVOICES" to store the program changes.

VERSION 3.0

Make the following changes to the program PROCESS.

Line:
```
45 W#=ABS(N#)*1ØØ+.1:X=W#/D1#:W#=W#-X*D1#
:V$=CHR$(X-(N#<Ø)*128):X=W#/D2#:W#=W#-X*D2#
:V$=V$+CHR$(X):X=W#/D3#:W#=W#-X*D3#
:V$=V$+CHR$(X)+CHR$(W#):RETURN
```

Line:
```
151 PRINT:PRINTCHR$(3Ø);" UNEXPECTED ERROR CODE ";
ERR/2+1;"IN LINE";ERL:GOSUB65:END
```

SAVE"PROCESS" to store the program changes.
Make the following changes to the program CHECKS.

Line:
```
89 W#=ABS(N#)*1ØØ+.1:X=W#/D1#:W#=W#-X*D1#
:V$=CHR$(X-(N#<Ø)*128):X=W#/D2#:W#=W#-X*D2#
:V$=V$+CHR$(X:X=W#/D3:W#=W#-X*D3#
:V$=V$+CHR$(X)+CHR$(W#:RETURN
```

Line:
```
315 PRINT:PRINTCHR$(3Ø);" UNEXPECTED ERROR CODE ";
ERR/2+1;"IN LINE";ERL:GOSUB141:END
```

SAVE"CHECKS" to store the program changes.
Make the following corrections to the program POST.

Line:
```
59 W#=ABS(N#)*1ØØ+.1:X=W#/D1#:W#=W#-X*D1#
:V$=CHR$(X-(N#<Ø)*128):X=W#/D2#:W#=W#-X*D2#
:V$=V$+CHR$(X):X=W#/D3#:W#=W#-X*D3#
:V$=V$+CHR$(X)+CHR$(W#):RETURN
```

Line:
```
195 PRINT:PRINTCHR$(3Ø);" UNEXPECTED ERROR CODE ";
ERR/2+1;"IN LINE";ERL:GOSUB1Ø7:END
```

SAVE"POST" to store the program changes.
Make the following changes to the program INVOICES.

Line:
```
79 N#=VAL(IN$):W#=ABS(N#)*1ØØ+.1:X=W#/D1#
:W#=W#-X*D1#:V$=CHR$(X-(N#<Ø)*128):X=W#/D2#
:W#=W#-X*D2#:V$=V$+CHR$(X):X=W#/D3#:W#=
W#-X*D3#:V$=V$+CHR$(X)+CHR$(W#):RETURN
```

Line:
```
3Ø5 PRINT:PRINTCHR$(3Ø);" UNEXPECTED ERROR CODE ";
ERR/2+1;"IN LINE";ERL:GOSUB133:END
```

SAVE"INVOICES" to store the program changes.

3—The General Ledger code numbers cannot be modified in SETUP. (Models I/III Version 3.1)

Make the following change to the program SETUP.

Line:
```
43 GOSUB89:PRINT@47Ø,"ALL CHANGES STORED "
:ONRF+1GOSUB199:GOTO2Ø7
```

SAVE"SETUP" to store the program changes.

4—Adjustments are not handled correctly in Select/Hold by vendor. Make the following correction to the program APS. (Model III Version 3.1)

Line:
```
299 GOSUB263:GET1,KR:PRINT@615,L2$;I1$;:
IA=(IV$=LEFT$(I1$,LEN(IV$))ANDI1$<>
"ADJUST")OR(IAANDI1$="ADJUST"):
IFIATHENGOSUB313
```

SAVE"APS" to store the change in the program.

ACCOUNTS RECEIVABLE (26-1555)

Following are four sets of program changes to correct separate problems in Accounts Receivable.

Correction 1—If any of the following symptoms occur, the Accounts Receivable index has been crashed, and the following Recovery program should be run. These corrections replace those previously published in March 1982. Model I Version 3.0)

Unexpected Error Code 64, Line 2850 ("ARS")
Unexpected Error Code 64, Line 1430 ("ARS")
Bad Record Number on program line with "GET" or "PUT" statement
When sorting, the statement "duplicate account # 0 exists" appears or transactions appear under the wrong account - i.e. some transactions for account 3 are on account 20's bill, etc.

In order to correct these above errors please run the following program written to reconstruct "CUSINDEX" and "TRANSACT" files of Model I Accounts Receivable program.

Quicker routines can be written, but this was written with the customer in mind. This routine will not recapture report numbers nor session numbers. It is merely intended to aid in recapturing data files rendered inaccessible because the Customer Index or Transaction files have been damaged or scrambled.

* * * * *This is a last resort. All transactions must be checked afterwards and verified. This program will pick up all transactions on the disk, including deleted transactions and previous period transactions not yet over-written in the current period.

Enter the following program in BASIC.

```
1   ' *** RECOVER *** (C) 1981 TANDY CORP
2   ' *** RECOVERS POINTERS/INDEX FOR ACCTS.
    RECEIVABLE
3   ' PROGRAM DISK SHOULD BE IN DRIVE Ø, DATA DISK IN
    DRIVE 1 --- ON 3 DRIVE SYSTEMS, SECOND DATA DISK
    IN DRIVE 2.
4   ' RECOVER WILL RUN SORT AND RETURN TO ARS ON
    COMPLETION.
5   ' **** FOR VERSION PRIOR TO 3.Ø, LINE 21Ø MUST BE
    MODIFIED: CHANGE "JD*128ASD$" TO "JD*127ASD$" AND
    "5ASDE$" TO "4ASDE$"
1Ø  CLEAR5ØØ:DEFINTA-Z:CLS:INPUT"PASSWORD";P$:INPUT"2
    OR 3 DRIVE SYSTEM";S
2Ø  GOSUB22Ø:OPEN"R",1,PT$:NT=LOF(1)*8:OPEN"R",2,
    PD$:NA=LOF(2)*2
3Ø  DIMIR(NT,2),AC(NA,2):CLS:PRINT@154,"FIRST
    PASS":PRINT@4Ø2,"WORKING ON TRANSACTION #"
4Ø  KK=NT:FORK=1TONT:GOSUB18Ø:PRINT@426,K:TR(K,Ø)=
    CVI(V1$):IFVØ$<"A" ORVØ$>"Z"THENKK=K:K=NT
5Ø  NEXTK:PRINT@154,"SECOND PASS":PRINT@426,"
    ":UN=KK:UT=KK-1:U=UT:
    FORK=1TOUT:IFTR(K,Ø)<ØTHENTR(K,Ø)=-TR(K,Ø):U=U-1
6Ø  PRINT@426,K:NEXT:CLS:PRINT@154,"FIRST
    PASS":PRINT@4Ø4,"WORKING ON ACCOUNT #"
7Ø  FORJ=1TONA:GOSUB2ØØ:PRINT@424,J:AC(J,Ø)=CVI(DD$)
    :IFAC(J,Ø)<1THENF=F+1 ELSECV#=CV#+CVD(DB$)
8Ø  NEXT:TN=J-1:TI=TN-F:JJ=J-1:PRINT@154,"SECOND
    PASS":PRINT@424,"     ":UX=UT+(TR(UT,Ø)=1)
9Ø  FORJ=1TOJJ:PRINT@424,J:PT=Ø:FT=Ø:AC=AC(J,Ø)
    :FORK=1TOUX:IFTR(K,Ø)=ACTHENTR(PT,2)=K
    :TR(K,1)=PT:PT=K:IFFT=ØTHENFT=K
1ØØ NEXTK:AC(J,1)=FT:AC(J,2)=PT:NEXTJ:CLS
    :PRINT@4Ø2,"RESETTING TRANSACTION POINTERS"
11Ø FORK=1TOUT:GOSUB18Ø:LSETV8$=MKI$(TR(K,1))
    :LSETV9$=MKI$(TR(K,2)):PUT1,KR:NEXT:CLOSE1
12Ø CLS:PRINT@4Ø4,"RESETTING ACCOUNT POINTERS"
13Ø FORJ=1TOTN:GOSUB2ØØ:LSETDI$=MKI$(AC(J,1))
    :LSETDJ$=MKI$(AC(J,2)):PUT2,JR:NEXT:CLOSE2
14Ø CLS:PRINT@41Ø,"WRITING INDEX":OPEN"O",1,
    "CUSINDEX."+P$+":1":PRINT#1,TI;TN;F;U;UT
    ;UN;Ø;1;1;1;1;1;1;1;1;STR$(CV#)+"DØ"
15Ø FORN=1TOTN-1:CN=N:IFAC(N,Ø)<1THENCN=-CN
16Ø PRINT#1,CN;CN;CN:NEXT:CLOSE1:OPEN"O",1,
    "TRANSFER:1":PRINT#1,PI$:PRINT#1,PD$
    :PRINT#1,PS$:PRINT#1,PT$:PRINT#1,PG$
    :CLOSE
17Ø CLS:PRINT@41Ø,"SORTING DATA":CLEAR5Ø:RUN"ARSORT""
18Ø KR=INT((K-1)/8)+1:KD=K-8*INT((K-1)/8)-1
19Ø FIELD1,KD*31ASVV$,1ASVØ$,23ASV1$,2ASV8$,2ASV9$
    ,3ASVX$:GET1,KR:RETURN
2ØØ JR=INT((J-1)/2)+1:JD=J-2*INT((J-1)/2)-1
21Ø FIELD2,JD*128ASD$,1Ø5ASDØ$,8ASDB$,4ASDC$,
    2ASDD$,5ASDE$,2ASDI$,2ASDJ$:GET2,JR:RETURN
```

```
220 PI$="TRANSACT."+P$+":"+CHR$(47+S):P$=P$+
    ":1":PI$="CUSINDEX."+P$:PD$="CUSDATA."+P$
    :PS$="CUSSETUP."+P$:PG$="GLFILE."+P$
    :RETURN
```

Type SAVE"RECOVERY" to save the above program for later use. Type RUN and press ⟨**ENTER**⟩ to start the recovery process. If the program is ever needed again load it from the disk with LOAD"RECOVERY"

Correction 2—When a correction entry is made to a payment, the correction entry replaces the previous payment on the statement. (Models I/III Versions 3.0 and Prior)

Make the following correction to line 2020 in version 3.0 of the program ARS. In versions prior to 3.0, this is line 1650.

Line:
```
2020 PR#=PR#-VE#:CB#=CB#-VE#:CV#=CV#-VE#
    :G#(VJ)=G#(VJ)-VE#:G#(2)=G#(2)-VE#
```

SAVE"ARS" to store the changes in the program.

Correction 3—These changes correct the following problems in versions prior to and including 3.1.:

Only one account can be deleted before posting. Accounts with transactions cannot be deleted immediately after posting.

The changes are made to the program ARS.

VERSIONS PRIOR TO 3.0

Line:
```
1300 IFDI<>0ANDUD<>0THENPRINT@832,CHR$(31);"CANNOT T
    DELETE THIS ACCOUNT UNTIL AFTER
    POSTING":GOSUB3340:GOTO1210ELSEIFPT=1THEN
    GOSUB3140
```

Line:
```
1990 U=0:UD=0:LPRINTLF$:LPRINT"END OF
    REPORT":LPRINTLL$:LPRINTCHR$(12):CLOSE3:GOTO16300
```

Line:
```
3410 FOR N=1 TO TN-1:INPUT#1,P(N,0),P(N,1),P(N,2)
    :NEXT:CLOSE1:UD=U
```

VERSION 3.0

Line:
```
1060 IFDI<>0ANDUD<>0THENPRINT@832,CHR$(31);"CANNOT
    DELETE THIS ACCOUNT UNTIL AFTER POSTING"
    :GOSUB2600:GOTO9600ELSEIFPT=1THENGOSUB2450
```

Line:
```
2640 OPEN"I",1,PI$:INPUT#1,TI,TN,F,U,UT,UN,EP,R0,
    R1,R2,R3, R4,R5,R6,R7,CV#:UD=U
```

Line:
```
1630 U=0:UD=0:LPRINT" ":LPRINT"END OF REPORT"
    :LPRINTLL$:GOSUB2515:CLOSE3:GOTO1360
```

VERSION 3.1

Line:
```
1630 U=0:UD=0:LPRINT" ":LPRINT"END OF REPORT"
    :LPRINTLL$:GOSUB2515:CLOSE3:GOTO1360
```

SAVE"ARS" to store the program changes.

Correction 4—Transactions cannot be entered or edited after statements are approved. (Versions 3.1 and prior)

Make the following corrections to the program ARS.

VERSIONS 3.0 & 3.1

Line:
```
1310 CLS:IFEP<3THENPRINT@474,"TRANSACTIONS":GOSUB240:
    RUN"ENTRAN"ELSEPRINT@462,"END OF PERIOD
    PROCESSING INCOMPLETE":GOSUB2600:GOTO230
```

VERSIONS PRIOR TO 3.0

Line:
```
1580 CLS:IFEP<3THENPRINT@474,"TRANSACTIONS":GOSUB340
    :CLEAR50:RUN"ENTRAN":END:ELSEPRINT@462,"END OF
    PERIOD PROCESSING INCOMPLETE":GOSUB3340:GOTO330
```

Type SAVE"ARS" ⟨**ENTER**⟩ to store the changes in the program.

DISK PAYROLL (26-1556)

In the following material is information regarding the way that Payroll calculates FICA and two corrections to existing program problems.

1—The way that Disk Payroll calculates FICA is correct. The Federal Government says FICA should be calculated by the following formula: GROSS *.0613. (Model I Non EIC Version)

Consider this Table:

Employee 1-Pay × .0613 equals Subtotal 555.53 × .0613 = 34.05
Employee 2-Pay × .0613 equals Subtotal 449.50 × .0613 = 27.55
Employee 3-Pay × .0613 equals Subtotal 99.63 × .0613 = 6.11
Employee 4-Pay × .0613 equals Subtotal 444.56 × .0613 = 27.25
Employee 5-Pay × .0613 equals Subtotal 999.50 × .0613 = 61.27

Gross Payroll Times .0613 equals Total or 2548.72 × .0613 = 156.24, BUT the Total of Each Employees FICA does not equal 156.24. It equals 156.23.

There are two ways of calculating FICA. One is to take a total sum of the gross wages of the employee and multiply it times .0613 or multiply the GROSS of each paycheck times .0613, then add those totals up to come up with the Tax. These two totals probably will not be equal due to the rounding errors involved in calculating the tax two different ways. When the employer fills out his forms, FICA is not calculated by multiplying 2549.18 × .0613 but by multiplying 2549.18 × .1226 to get the amount that he owes to the government.

2—If the weeks worked, workmen's comp, or net pay in the employee's personnel information section cannot be changed, make the following change to the program PR4ADD. (Model I Versions Prior to 2.0)

Line:
```
3900 FORI=Z1TOZ2:MID$(G$,I*8-7,8)=MKD$(E#(I-Z3))
    :NEXTI:MID$(G$,Z2*8-7,8)=MKD$(E#(17))
    :LSETGG$=G$:LSETFF$=N$:PUT3,N:PUT4,1+2*(N-1)
    +INT((Z-2)/2):GOTO810
```

SAVE"PR4ADD" to store the program changes.

3—Two new steps have been added to the calculation process for Minnesota State Tax Tables. The additions involve a 7% surcharge. The necessary program changes can be implemented by making the following changes to the program STATETAX. (Models I/III Version 2.0)

Line:
```
7900 PRINT#1,"5900E#(I)=((E#(I)-CVS(MID$(N$,111,4)))
    *1.084)/K#:IFE#(I)<.01THENE#(I)=0:GOTO6500ELSE
    6500":CLOSE:GOTO2000
```

ADD the following line:

Line:
```
3160 PRINT#1,"5115G#=G#*.909"
```

SAVE"STATETAX" to store the changes in the program.

Then RUN "STATETAX" and reenter the tax tables using the instructions in the manual.

MEDICAL OFFICE SYSTEM (26-1568)

Following are two sets of corrections for this program:

Correction 1—When option 2 of statement printing is selected, zero balance accounts will not be printed. Make the following corrections to the program BLLPRT. (Model I/III Version 1.0)

Line:
```
1100 PRINT@728,"SELECT OPTION: ";:FL=-1
    :GOSUB210:Q=VAL(IN$):PRINT:IFQ<1ORQ>3
    THEN1100ELSEPF=Q=1:X0=Q<>2:IFQ=1THEN
    POKE16424,43ELSEPOKE16424,67
```

Line:
```
1130 IFA=AHTHEN1140ELSEIFX0AND(BA=0ORI$="N")
    THEN1210ELSEGOSUB1215:IFPFTHENLPRINTLF$
    :LPRINTLF$:LPRINTLF$:IFBA<0THENLPRINT
    TAB(40)"CREDIT BALANCE"ELSELPRINTTAB(42)
    "PLEASE REMIT"
```

Line:
```
1140 IFX0AND(BA=0ORI$="N")THENGOSUB1250:
    GOTO1210ELSEGOSUB1215
```

SAVE"BLLPRT" to store the program changes.

Correction 2—Medical Office System contains a bug which may cause problems if more than three patient data files are used.

**Changes to MOS are somewhat critical, the following process should be followed closely.

1. In BASIC, LOAD"ALPHASOR".
2. Type these lines as shown.

Line:
```
1240 FZ=INT((AC-1)/Z1):IFFZ<>FOTHENFO=FZ:CLOSE3:
     FM$="PDATA"+CHR$(48+FO)+":3":NU=3:GOSUB400 :CLS

1250 LR=AC-FO*Z1:PR=INT((LR-1)/2)+1:SR=LR-2*
     INT((LR-1)/2)-1
```

3. SAVE"ALPHASOR"
4. In BASIC, LOAD"MENU".
 Add these lines as shown.

```
120 FZ=INT((AC-1)/Z1):IFFZ<>FOTHENFO=FZ:CLOSE3:
    FM$="PDATA"+CHR$(48+FO)+":3":NU=3:GOSUB400
140 LR=AC-FO*Z1:PR=INT((LR-1)/2)+1:SR=LR-2*
    INT((LR-1)/2)-1
1400 DEFFNA(X)=PEEK(X)+PEEK(X+1)*256:E=FNA(&H40A4)
1410 S=FNA(E):IFFNA(S+2)<1000THENE=S:GOTO1410
1420 PRINT"TP=TP+"S-FNA(&H40A4)"IN LINE 1110":END
```

5. TYPE: GOTO1400

(The computer will pause and print 'TP = TP + NNNN in line 1110. The NNNN will be a number.)

6. Type LIST 1110 ⟨**ENTER**⟩

(If the number after TP + TP + . . . does not match the computer generated number of step 5, the line must be edited to contain the new value.)

7. Type EDIT 1170 ⟨**ENTER**⟩

Line 1170 should look like this:

```
1170 DATAMONSORT,PRINT STATEMENTS,BLLPRT,SET UP
     SYSTEM DISKS,INIT,16922,16923,16924,1390,4170,3,
     16452,16453,16454,660,3960,6
```

(If the numbers that appear after 16924 are not 1390,4170, edit them to conform to the new line 1170. There are no other changes in the line.)

8. Type SAVE"MENU"⟨**ENTER**⟩

**FORMAT A NEW DATA DISK FOR EACH PATIENT DATA DISK.

Type in this PROGRAM:

```
10 CLEAR2000:CLS:AC=1:FO=-1:B2=-1
20 IFPEEK(293)=73THENZ1=1390ELSEZ1=660
30 D$(1)="OLD":D$(2)="NEW"
40 CLS:INPUT"WHAT IS YOUR PASSWORD? (PRESS <ENTER> IF F
   NONE)";PW$
50 IFPW$=""THENPW$="PASSWORD"
60 PW$="."+PW$
70 IFFO<>INT(AC/(Z1+1))THENCLOSE1:FO=
   INT(AC/(Z1+1)):D=1:DN=FO+1:GOSUB140:OPEN"R",1,
   "PDATA"+CHR$(FO+48)+PW$+":1"
80 PRINT@339,AC
90 LR=AC-FO*Z1+1:PR=INT((LR-1)/2)+1:SR=LR-2*
   INT((LR-1)/2)-1
100 FIELD1,SR*127ASA1$,127ASA$:GET1,PR
110 B1=INT((AC-1)/Z1):IFB1<>B2THENB2=B1:CLOSE2
   :FM$="PDATA"+CHR$(48+B1)+":2":D=2:DN=B2+1:GOSUB14 4
   0:OPEN"R",2,"PDATA"+CHR$(48+B1)+PW$+":2"
120 B=AC-B1*Z1:B3=INT((B-1)/2)+1:B4=B-2*
   INT((B-1)/2)-1
130 FIELD2,B4*127ASB1$,127ASB$:LSETB$=A$
   :PUT2,B3:AC=AC+1:GOTO70
140 IFAC>1THENMS$="NEXT"
150 CLS:PRINT"PUT "D$(D)" PATIENT DATA DISK"DN"IN
   DRIVE"D
160 PRINT:PRINT"PRESS <ENTER> WHEN READY, OR <@> TO
   END"
170 W$=INKEY$:IFW$=""THEN170
180 IFW$=CHR$(13)THENCLS:PRINT@320,"WORKING ON RECORD D
   #";:RETURN
190 IFW$="@"THENCLOSE:ENDELSE170
```

FOLLOW THE PROMPTS!

Insert old and new disks as requested. These new disks will be used with the modified MOS disk. DO NOT attempt to use the old patient data disks with the modified program.

REAL ESTATE VOL. III (26-1573)

If the "End of Period Payment" is used, the first factor in the factor column on the hardcopy option is deleted and all factors are moved up by one. The screen display is all right. (Models I/III Version 3.0)

Correct the problem by CLOADing the Variable Income Analysis tape and making the following change.

Line:
```
6070 IFJ=1THENB=(1/(1+(B/N1)))^(I-1):GOTO6090
```

CSAVE"V" to save the changes on a new cassette.

NOTE On the Model III the "up-arrow" key appears as a left bracket on the screen, but it functions as an exponentiation symbol in the program.

RUNCOBOL

COBOL (26-2203 Model III Version 1.3b)
COBOL RUNTIME (26-2206 Model I Version 1.3b)
COBOL RUNTIME (26-2207 Model III Version 1.3b)

The following problem exists in RUNCOBOL of all three of the above programs.

The OPEN EXTEND OPTION is not functional in the RUNCOBOL runtime program. An ERROR 30 message results when OPEN EXTEND is used with a file that has been previously opened and closed.

Below is a program which when run will result in an error 30.

```
000100 IDENTIFICATION DIVISION.
000110 PROGRAM-ID.                    ERROR-30.
000120 ENVIORNMENT DIVISION.
000130 CONFIGURATION SECTION.
000140 SOURCE-COMPUTER.                MODEL-III-48K.
000150 OBJECT-COMPUTER.                MODEL-III-48K.
000160 INPUT-OUTPUT SECTION.
000170 FILE-CONTROL.
000180     SELECT I-O-FILE
000190         ASSIGN TO RANDOM,  "ERROR30/DAT";
000191         ORGANIZATION IS SEQUENTIAL;
000192         ACCESS IS SEQUENTIAL.
000200 DATA DIVISION.
000210 FILE SECTION.
000220 FD  I-O-FILE
000230         RECORD CONTAINS 32 CHARACTERS
000240         DATA RECORD IS I-O-RECORD.
000250 01  I-O-RECORD.
000260     02  DATA-RECORD            PICTURE  X(032).
000270 PROCEDURE DIVISION.
000280 BEGIN.
000290     DISPLAY "FIRST WRITE",
000300         LINE 01, POSITION 20, ERASE.
000310     OPEN OUTPUT I-O-FILE.
000320     MOVE SPACES TO DATA-RECORD
000330     WRITE I-O-RECORD.
000340     CLOSE I-O-FILE.
000350
000351     DISPLAY "SECOND WRITE",
000352         LINE 02, POSITION 20, ERASE.
000360     OPEN EXTEND I-O-FILE.
000370     MOVE SPACES TO DATA-RECORD.
000380     WRITE I-O-RECORD.
000390     CLOSE I-O-FILE.
000400
000410     STOP RUN.
EOF
```

Applying the following patches at TRSDOS READY will correct this problem for all of the above packages.

```
PATCH RUNCOBOL/CMD (ADD=9A2A,FIND=000000000000,
                              CHG=CDF0AAC0FDCB
PATCH RUNCOBOL/CMD (ADD=9A30,FIND=000000000000,
                              CHG=0266C8FDE5E1
PATCH RUNCOBOL/CMD (ADD=9A36,FIND=000000000000,
                              CHG=111C0019EBDD
PATCH RUNCOBOL/CMD (ADD=9A3C,FIND=00000000,
                              CHG=E5FDE5CD
PATCH RUNCOBOL/CMD (ADD=9A40,FIND=000000000000,
                             CHG=4844FDE1DDE1
```

```
PATCH RUNCOBOL/CMD (ADD=9A46,FIND=ØØØØØØØØØØØØ,
                    CHG=C9FDCBØ266CA
PATCH RUNCOBOL/CMD (ADD=9A4C,FIND=ØØØØØØØØ,
                    CHG=569AE5D5
PATCH RUNCOBOL/CMD (ADD=9A5Ø,FIND=ØØØØØØØØØØØØ,
                    CHG=CDF7ADD1E1C8
PATCH RUNCOBOL/CMD (ADD=9A56,FIND=ØØØØØØØØ,
                    CHG=CDEBADC9
PATCH RUNCOBOL/CMD (ADD=A43D,FIND=CDFØAA,
                    CHG=CD2A9A
PATCH RUNCOBOL/CMD (ADD=A7CF,FIND=FD,CHG=C3
PATCH RUNCOBOL/CMD (ADD=A7DØ,FIND=CBØ266,CHG=DEA7ØØ
PATCH RUNCOBOL/CMD (ADD=ABØ5,FIND=CDEBAD,CHG=CD479A
```

COBOL COMPILER (26-2203)

The following correction was published in the May issue of TRS-80 Microcomputer News but the last two patches were missing, and the change (CHG) string was missing on the third from the last patch. (Model I/III Version 1.3b)

When RUNCOBOL (MOD I/III) attempts to call an independent segment overlay, it will have a LOAD FAIL ERROR message and return to DOS. RUNCOBOL is unable to load and execute the machine language program (independent segment) and is forced to return to TRSDOS.

The following PATCHes will correct the above problem:

```
PATCH RUNCOBOL/CMD (ADD=9DB8,FIND=CD11AEC2419F,
                    CHG=ØØØØØØØØØØØØ)
PATCH RUNCOBOL/CMD (ADD=9DC4,FIND=2A,CHG=21)
PATCH RUNCOBOL/CMD (ADD=ADØD,FIND=19,CHG=Ø9)
PATCH RUNCOBOL/CMD (ADD=ACFE,FIND=D6ØA,CHG=C318)
PATCH RUNCOBOL/CMD (ADD=ADØØ,FIND=3Ø,CHG=9A)
PATCH RUNCOBOL/CMD (ADD=9A18,FIND=ØØØØØØØØ,
                    CHG=FEØADAØ4)
PATCH RUNCOBOL/CMD (ADD=9A1C,FIND=ØØØØØØØØ,
                    CHG=ADD611DA)
PATCH RUNCOBOL/CMD (ADD=9A2Ø,FIND=ØØØØØØØØ,
                    CHG=11ADFEØ6)
PATCH RUNCOBOL/CMD (ADD=9A24,FIND=ØØØØØØØØØØØØ,
                    CHG=D211ADC3Ø2AD)
PATCH RUNCOBOL/CMD (ADD=9B65,FIND=2A69AE4E23,
                    CHG=46ED43D1AF)
PATCH RUNCOBOL/CMD (ADD=9B6A,FIND=46ED43D1AF,
                    CHG=CD8CA82Ø5C)
PATCH RUNCOBOL/CMD (ADD=9DCB,FIND=489F,CHG=D49D)
PATCH RUNCOBOL/CMD (ADD=9B6Ø,FIND=CD8CA82Ø66,
                    CHG=2A69AE4E23)
```

Some Straight Talk About the Radio Shack Network Systems

Radio Shack's Network 2 and Network 3 systems add convenience to classroom computing. The following facts and figures have been prepared to show that Networking with the TRS-80 can be economical, too.

The Network 2 system keeps control of loading programs in the hands of the teacher. Up to sixteen student stations can be loaded with the same program simultaneously, or different programs can be loaded into different stations. Here is a summary of Network 2 costs for a 16-station network with a 48K two-disk host:

Radio Shack TRS-80 Network 2 System
Optimum Configuration = 16 Student Stations

Equipment:	
16 16K Model III TRS-80s	$15,984.00
1 Radio Shack Network 2 Controller	$499.00
1 TRS-80 48K two-drive disk system	$2,295.00
	$18,778.00
Cost per year for use over a 5-year period (excluding service contract/maintenance)	$3,755.60
Cost per student station per year	$234.73
Cost per student hour (1000 hours per station per year)	$.23

Network 3 gives non-disk student stations most of the capabilities of a disk system. The system lets students sitting at the student stations load and save programs using the host disk. A printer at the host computer can be shared by the student stations. With 32K or 48K RAM in the student stations, many Radio Shack educational software programs will run on the Network 3. (Business programs or programs in compiler languages will NOT run on the Network 3.) The following is a summary of costs for a Network 3 system in which each student station has 48K RAM:

Radio Shack TRS-80 Network 3 System
Maximum Configuration = 16 Student Stations

Student Stations:	
16 48K Model III TRS-80 tape systems (includes 16 16K Model IIIs @999.00, 32 16K RAM kits @49.00, installation fees @15.00 per RAM kit, 16 RS-232 boards @99.00 and installation fees @17.50 per board)	$19,896.00
Network 3 Controller, Operating System, and Cables (typial cable configuration)	$1,614.00
Host Computer: 48K Model III dual disk system	$2,295.00
Network 2 Controller (used for simultaneous host-to-student-station program transfer and loading Network 3 communications software)	$499.00
Total Network 3 System	$24,304.40
Cost per year for use over a 5-year period (excluding service contract/maintenance)	$4,860.88
Cost per student station per year	$303.81
Cost per student hour (1000 hours per station per year)	$.30

Note that the Network 3 system cost is less than that for 16 Radio Shack disk systems with 48K:

16 48K single-disk student stations (each $1849.00)	$29,584.00
Cost per year for use over a 5-year period (excluding service contract/maintenance)	$5,916.80
Cost per student station per year	$369.80
Cost per student hour (1000 hours per station per year)	$.37

News From The Education Division

The Radio Shack Education Division has grown dramatically both in personnel and in the number of different types of programs we're producing, in the past year alone. Because our commitment to the educational user is a source of pride at Radio Shack, we'd like to give you an overview in this education issue of the exciting things we've been doing to support educational use of the TRS-80 microcomputer.

Although it isn't generally known, many microcomputer manufacturers don't develop courseware for their own machines. Instead, they rely on outside developers. Radio Shack was one of the first manufacturers to see what the microcomputer could do for education and to get involved in courseware development. Students at all grade levels and in all subjects are benefiting from the extra practice, instruction, or exploration available from the computer. With good courseware, the computer is easy to use and nonjudgmental. It is allowing students to work at their own pace, free to learn from their mistakes without embarrassment and free to do their own best work without peer pressure. Best of all, the computer has unending patience, rewards students with positive feedback messages, and makes students view problem-solving as a game. From the start of the Radio Shack Education Division two and a half years ago, our purpose has been to develop programs that are constructive in all of these ways.

The first program we released was the Radio Shack K-8 Math Program, Volume One. K-8 Math is a classic drill-and-practice CAI (Computer Assisted Instruction) program. In this case, the program provides varied and intensive skill-building work by randomly generating problems in numeration, addition, subtraction, multiplication, and division for grades K through 8. Field tests prove that K-8 Math works. By motivating, encouraging, and drilling students on math basic skills, K-8 Math has been effective in raising math competency levels. The quality of K-8 Math has become a standard for other Radio Shack courseware programs.

Since K-8 Math, we've expanded our product development efforts. Many more kinds of CAI programs are now being released, including tutorial programs, simulation and problem solving programs, and basic skills practice programs in other areas (like reading and business). Many of our latest products have been featured in recent TRS-80 MICROCOMPUTER NEWS articles. (You'll also find descriptions of many in this month's Computer Customer Service article.) In addition to the area of CAI, our current product development efforts are in these categories: (1) computer literacy, (2) computer managed instruction, (3) authoring systems, (4) resource materials, and (5) special hardware.

COMPUTER LITERACY

Computer literacy is a fairly new subject area in many schools. On one level, computer literacy deals with the history of computers and their place in society—roughly comparable to teaching about the Wright brothers and the history of air and space travel. To teach students about computers is to teach them about something that will influence their lives, directly or indirectly. Radio Shack's Basic Computer Literacy package COMPUTERS PAST AND PRESENT is a complete teaching package for this kind of computer literacy course.

On another level, computer literacy deals with the ability to use, and perhaps to program, the computer. This area can emphasize mainly the practical skills, as when word processing is taught as part of a vocational program. Or it can get into programming and the realm of intellectual problem-solving, with the computer as laboratory. Radio Shack's COMPUTER EDUCATION SERIES helps meet the need for this kind of computer literacy with three complete teaching packages designed for secondary and post-secondary students. Taken consecutively, these courses lead students and teachers from a first introduction to the computer through advanced programming in the BASIC computer language.

The COMPUTER EDUCATION SERIES is designed so it can be used by teachers who may not know a lot about computers or programming. This addresses a problem we see frequently in the schools: wanting to offer some computer programming courses but not having teachers who are experienced in that field. More volumes in the series are planned.

Educators at all types of schools have affirmed the importance of computer literacy and the fact that the well-educated adult of the future will have to know something about computers. Elementary educators have recently spoken of computer literacy as "the fourth 'R'" (in addition to Reading, Writing, and Arithmetic). And Harvard University now requires all undergraduates to meet a "Quantitative Reasoning" requirement that includes "use of the computer". Harvard students usually meet part of the requirement by writing and running a short computer program. Clearly the computer in education is an idea whose time has come.

COMPUTER MANAGED INSTRUCTION

The computer can save today's overworked teacher some valuable time by providing some of the supplementary drill work that many students need so badly. Another time-saver is computer managed instruction. Several Radio Shack programs, including the K-8 MATH WITH STUDENT MANAGEMENT program, NUMERIC DATA ENTRY PRACTICE, C.A.R.D. I: SENTENCES, the STUDENT RECORDS SYSTEM for the HIGH MOTIVATION READING SERIES, and QUICK QUIZ: A MINI-AUTHORING SYSTEM, feature automatic storage of student scores on diskette. This saves time when the lesson is taken because the teacher can keep permanent records without having to record performance information off the screen when the student finishes the lesson. And it saves time later because the computer calculates the score—the teacher doesn't have to sit up all night grading quizzes and exercises.

AUTHORING SYSTEMS

The Radio Shack Education Division is busy providing authoring systems which enable teachers to design and create their own courseware lessons in any subject without having to have a knowledge of programming. We offer a series of these systems, appropriate for everything from test generation to development of a complete instructional series with record keeping, conditional branching, and student management. Examples are TRS-80 MicroPILOT™ and TRS-80 AUTHOR I.

RESOURCE MATERIALS FOR EDUCATORS

Educational Resource Materials has been an important area of recent emphasis for our Publications group. The RADIO SHACK PROPOSAL WRITING GUIDE is a detailed step-by-step guide to writing successful proposals for CAI funding. The brand new EDUCATIONAL SOFTWARE SOURCEBOOK lists many TRS-80 educational software products available from numerous companies and individuals.

To introduce educators to microcomputer applications, Radio Shack publishes the MICROCOMPUTER INFORMATION HANDBOOK FOR EDUCATORS. A reference source for classroom use is the RADIO SHACK K-8 MATH CROSS-REFERENCE, which matches lesson numbers from the Radio Shack K-8 Math Program with unit, chapter, and page numbers from six of the major mathematics basal series.

SPECIAL HARDWARE

Our special hardware for the classroom is designed to provide teachers and schools with the most convenient, cost-effective configurations possible. Our first educational hardware product, the NETWORK 1 CONTROLLER, allowed up to 16 TRS-80 cassette-based student stations to be connected to a single host TRS-80 disk system, thereby allowing the stations to share the disk and printer on the host. The more recent NETWORK 2 CONTROLLER replaces the original Network 1 and supports the higher communication rate possible with the Model III, while still retaining compatibility with the many Model I's in schools.

The new NETWORK 3 is a polled system that performs different networking functions from the Network 2. Among the new features of Network 3 is the ability to use student management programs with the Network configuration.

WHO IS THE EDUCATION DIVISION ?

Supporting educational applications of the TRS-80 microcomputer to help YOUR school's microcomputer ideas become reality is the purpose of the Radio Shack Education Division. The Education Division includes six different areas: (1) Educational Product Development (including the areas of Software and Publications), (2) Educational Consultants, (3) Educational Sales, (4) Publisher Relations, (5) Industrial Training, and (6) National Bid.

The Educational Product Development Department produces the instructionally sound, effective, and properly validated microcomputer-based education materials that we market. The number of writers and programmers in this department has more than doubled in the past year. This department is divided into the two areas of Software Development and Publications.

The Software Development area uses in-house programmers and works with contract programmers to develop the computer programs in Radio Shack's educational courseware packages. Operation and content of our software are carefully polished through testing, revision, and retesting. Programmers, testers, and writers give special attention to user-friendliness, making sure that program options are clearly defined, that the program is easy to use, and that the student is constantly reinforced with positive messages or with guiding hints and corrections. Radio Shack Consultants, experts in various fields of education, are called upon to evaluate and polish program curricula. Screen displays are scrutinized for neatness, accuracy, and clarity.

Once the software is ready, the Publications staff writes and edits the user's manuals, which are essential parts of the courseware packages. Traditionally, a "walk-through" or hands-on demonstration of the program is at the core of Education Division manuals to provide an easy introduction to the program's main features. Additional features of the program may be outlined in a "reference section." Where appropriate, a summary of the content of each lesson is included. Our goal is to insure that our courseware content, manual text, binders, artwork, and so on, measure up to the textbook standards used by other educational publishers.

In addition to working on user's manuals, Publications staff produce workshop and training materials and educational resources for classroom and administrative use.

Another important resource is our Educational Consultants. These consultants provide expert guidance in curriculum design and other education-related issues. Among our consultants are experts in reading and mathematics/science instruction, district-wide computer applications, courseware design and the use of authoring systems, grants and federal funding, and computer literacy instruction. University professors, former school district superintendents, and other educators experienced both in teaching and administration are among our consultants. They work to help us keep the educator's viewpoint constantly in sight.

Radio Shack Regional Educational Sales Coordinators are the Educational Sales personnel in the field. These are the people who support Radio Shack stores in helping schools and districts determine how to meet specific educational computing needs. Educational Sales has grown rapidly, from 9 full-time Regional Coordinators this time last year to a current team of 21, located around the country.

These Regional Coordinators know that before a microcomputer purchase is made, schools have to spend a lot of time in planning their applications, comparing configurations and costs, and providing inservice training for the teachers who will be using these systems in their classrooms. Therefore, a major part of the Coordinator's job is serving as resource person for school districts. They frequently give workshops and inservice training sessions on computer use, they work to help schools and districts find answers to their particular microcomputer needs, and they help educational users get in touch with other Radio Shack support people in the area.

The people working in the Publisher Relations area have the important job of maintaining cooperative relationships between the Education Division and major educational textbook publishers. By encouraging educational publishers to produce electronic learning materials for the TRS-80, we're

helping to increase the applications available to you, the user. No microcomputer manufacturer can produce the software to meet every need. Established educational publishers will be an important source of educational software in the years ahead.

Just a few of the publishers the Education Division has cooperative agreements with include South-Western Publishing Co., A Division of SFN; Educational Testing Service of Princeton, N.J.; Random House School Division; Psychological Corporation, A Division of Harcourt Brace Jovanovich; and Harper and Row. By agreements like these, Radio Shack provides marketing, sales, and technical support for courseware development by educational publishers.

Another area, Industrial Training, will become increasingly important in the coming years. Computer technology offers great potential in this kind of post-secondary education, and Radio Shack is currently working to develop such programs. Authoring systems will be especially important in creating training materials for industires of all types.

The final area, National Bid, was established to provide fast, accurate responses to all of the bid invitations and requests for price quotation that Radio Shack receives from educational institutions. Through National Bid, schools can be offered special prices on Radio Shack hardware and software, either in quantity or under the terms of numerous purchase agreements with state education agencies and major school districts.

OUTLOOK: SPECIAL PROJECTS AND PRODUCTS

Now let's look at some of the special projects and products that the Education Division is currently involved in.

CHILDREN'S TELEVISION WORKSHOP SOFTWARE

One exciting new development announced by the Education Division this year is the agreement with Children's Television Workshop, the creators of Sesame Street and The Electric Company. Sometime over the next year, CHILDREN'S TELEVISION WORKSHOP will develop and release a series of ten game-style educational software packages for use exclusively with the TRS-80 Color Computer. Later, two instructional series will be developed and released. One will be a reading classroom package for grades 1-4, emphasizing visual discrimination, matching skills, and reading comprehension.

NEW EDUCATIONAL HARDWARE

In the area of hardware, Radio Shack this year added a third classroom network system to the catalog. The NETWORK 3 CONTROLLER, featured in a July TRS-80 MICROCOMPUTER NEWS article, brings new disk-based capabilities to the classroom. Network 2 provides a fast, convenient way for teachers to download instructional programs through a system of connected tape-based student stations. But now Network 3 gives these tape-based student stations many of the capabilities of individual disk systems, including the ability to run student management programs.

HOME INSTRUCTION SERIES

Radio Shack's HOME INSTRUCTION SERIES is a new courseware development that has just begun to become a reality. The series is designed to provide children with entertaining education through programs that are relatively inexpensive and run on the minimum 4K Color Computer tape system. Packages released so far emphasize language skills for children reading at about grade levels 4-6. Four READING IS FUN packages are now available (these are spin-offs of the classroom HIGH MOTIVATION READING SERIES). Two VOCABULARY TUTOR programs are soon to follow. Watch for many new additions to this promising series.

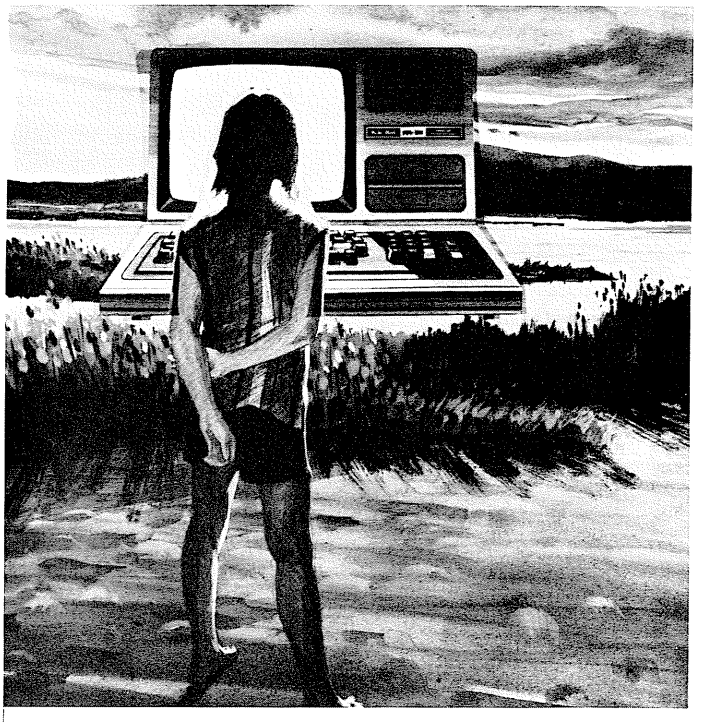
Color LOGO

Later this year, Radio Shack will release Color LOGO, a TRS-80 version of the well-known LOGO program. Color LOGO will run on a 32K disk-based or 16K tape-based TRS-80 Color Computer. Color LOGO features a "turtle" that moves around on the screen and leaves a trail of color in response to directions that are typed in at the keyboard. When children tell the computer how to move, they learn about direction, distance, and angle. More importantly, they have the chance to be creative.

Color LOGO allows you to choreograph fairly sophisticated turtle movements and save these mini-programs on tape or disk. Adults as well as children should find hours of fascination in this program.

AND IN THE FUTURE . . . ?

The products and activities you've just read about are only a few of the things that are new at the Radio Shack Education Division. Also coming forth will be a comprehensive series of Chemistry simulations for the TRS-80 Color Computer and for the TRS-80 Model I and Model III, a second volume of our popular K-8 Math Program, more Vocabulary Tutor programs for the Color Computer, more volumes in the Radio Shack Computer Education Series . . . and the list goes on and on. In the future, you can expect to see Radio Shack continue to produce more education-oriented microcomputer products than any other microcomputer manufacturer.



Programming Techniques

Auto FIELD Routine for Model II

Max Lupul
10448 Amigo Avenue
Northridge, CA 91326

I have numerous programs which use multiple random access files containing a hodge-podge of fields and I have gotten tired of writing lengthy codes to field these files. I have solved my problem with the enclosed 'all purpose file fielder' which uses a string 'A$' to contain all the specs for a specific file.

The first two characters in 'A$' contain the buffer number I wish to use, the balance of 'A$' contains two-character specifications of the field lengths. Thus line 10 in the listing specifies that buffer #1 will contain a field of 14 bytes, followed by one of 12, followed by 15 2-byte fields, 4 4-byte fields and one 35-byte field. Line 30 specifies that buffer #2 will contain a 22-byte, a 21-byte and a 2-byte field.

'L%' in the subroutine computes the length of the buffer; 'F%' computes the length of the masking dummy variable, while X$ is a double subscripted buffer variable, with the first subscript identifying the position in the buffer and the second the identity of the buffer. The DIM X$ statement should dimension X$ to the largest buffer size in the first subscript and to the number of buffers in the second subscript.

I am providing this with the thought that some other users might find this approach viable, or might want to modify it to suit his/her application.

```
5  CLEAR 10000
     : DIM X$(24, 2)
10 A$="011412020202020202020202020202020202020202040404
   040435"
20 GOSUB 5000
30 A$="02222102"
40 GOSUB 5000
50 STOP 'PROGRAM WOULD CONTINUE HERE
5000 L%=0
     : FOR I=3 TO LEN(A$) STEP 2
     : L%=L%+VAL(MID$(A$, I, 200
     : NEXT I
5010 OPEN "D", VAL(LEFT$(A$,2)), "TEST"+LEFT$(A$,
   2)+"/DAT", L%
5020 F%=0
     : FOR I=1 TO LEN(A$)/2-1
     : FIELD VAL(LEFT$(A$, 2)), F% AS D$,
     VAL(MID$(A$, I*2+1, 2)) AS X$(I,
     VAL(LEFT$(A$, 2)))
     : NEXT I
5030 RETURN
```

String Machine Language Programs

Christopher P. Hamkins
11½ Locust Street
Waterloo, NY 13165

I have discovered an interesting way to access user-defined machine language subroutines from BASIC on my Model II.

The object code is encoded in a string variable, and then a jump to the string contents is executed. This can be accomplished from BASIC.

String variables are stored in two parts, a 3-byte descriptor and the string itself. The descriptor has the form:

length	memory location	
1st byte	2nd	3rd bytes

From BASIC, I can get the address of the string descriptor using the 'VARPTR' function. This does not allow me to jump directly to the string, but I can use this to jump to the string descriptor location by defining DEFUSRn = VARPTR-(STRING$). If I arrange things so that the first byte of the descriptor is a "Jump" code, jumping to the descriptor location will execute a jump to the next two bytes, which are the location of the string. This allows a complete access to the machine language program without ever leaving BASIC, and makes the routine completely relocatable. The disadvantage is that absolute program addressing cannot be used.

I have used this technique to make a BASIC subroutine which invokes the "scroll" service call to protect the top of the screen from scrolling. The listing is enclosed.

P.S. The "Customer Service" section is by far the best and most informative. Please continue it and expand it.

```
110 GOSUB 230
120 'THIS PROGRAM USES THE SERVICE CALL NUMBER
     27--'SCROLL' TO PROTECT THE
130 'TOP OF THE SCREEN FROM SCROLLING ACTION.
     THIS IS DONE WITH A USER-
140 'DEFINED MACHINE LANGUAGE SUBROUTINE ENCODED
     IN THE STRING SC$.  THE
150 'TRANSFER TO THE SUBROUTINE IS ACCOMPLISHED
     BY SETTING THE LENGTH OF THE
160 'STRING TO 195, WHICH RESULTS IN A 'C3' CODE
     FOR THE FIRST BYTE OF THE
170 'STRING DESCRIPTOR.  SINCE THE 'C3' CODE IS
     THE 'JP pq' STATEMENT, AND
180 'THE NEXT TWO BYTES OF THE STRING DESCRIPTOR
     ARE ACTUAL MEMORY LOCATION
190 'OF THE STRING, JUMPING TO THE STRING
     DESCRIPTOR LOCATION WILL DO A
```

```
200 'JUMP TO THE BEGINNING OF THE ACTUAL STRING.
    THE MACHINE LANGUAGE
210 'SUBROUTINE IS ENCODED BYTE BY BYTE IN THE
    STRING BEFORE THE JUMP.
220 'THE PARAMETER LL IS THE NUMBER OF LINES TO
    BE PROTECTED.
230 INPUT LL%
240 IF (LL%<0) THEN LL%=0
250 IF (LL%>22)THEN LL%=22
260 SC$=CHR$(6)+ CHR$(LL%)+ CHR$(62)+ CHR$(27)+
    CHR$(207)+ CHR$(201)+ STRING$(189,CHR$(118))
270 L%=VARPTR(SC$)
280 DEFUSR0=L%
290 DUMMY=USR0(DUMMY)
300 RETURN
```

Calculator

Allen Banick, Jr.
2612 Redlands Drive
Costa Mesa, CA 92627

```
10 'BASIC CALCULATION PROGRAM THIS PROGRAM
    WRITTEN FOR THE TRS-80 MOD-II BY ALLEN
    BANICK, JR. 2612 REDLANDS DR. COSTA MESA CA.
      92627  THIS PROGRAM MAY BE PUBLISHED AS I
    CLAIM NO COPY RIGHTS
20 'THIS IS A SIMPLE ADDING MACHINE PROGRAM USING
    SOME OF THE SPECIAL FEATURES OF THE MOD-II.
30 'FUNCTIONS
    <F1>          =+        <F2>          =-
    <LEFT ARROW>=*          <RIGHT ARROW>=/
    <UP ARROW>  =SUB-TOT <DOWN ARROW> =TOTAL
    <E>           =END      <C>           =CLEAR
                                            SCREEN
    <P>           =PRT ON   <O>           =PRINT OFF
40 ' ADDITIONAL FUNCTIONS CAN EASILY BE ADDED
    WITHIN LINES 180-270
50 DEFDBL A-D
60 CLS
70 PRINT
    : A=0
    : F$=""
80 PRINT "F>";
90 F$=INKEY$
100 IF F$="" THEN 90
110 IF F$=CHR$(01) THEN F$="+"
120 IF F$=CHR$(02) THEN F$="-"
130 IF F$=CHR$(28) THEN F$="*"
140 IF F$=CHR$(29) THEN F$="/"
150 IF F$=CHR$(30) THEN F$="S"
160 IF F$=CHR$(31) THEN F$="T"
170 IF F$="+" THEN PRINT F$;
    : INPUT A
    : AS=AS+A
    : GOTO 70
180 IF F$="-" THEN PRINT F$;
    : INPUT A
    : AS=AS-A
    : GOTO 70
190 IF F$="*" THEN PRINT F$;
    : INPUT A
    : AS=AS*A
    : GOTO 70
200 IF F$="/" THEN PRINT F$;
    : INPUT A
    : AS=AS/A
    : GOTO 70
220 IF F$="S" THEN PRINT CHR$(26); "ST";
    CHR$(25); AS
    : GOTO 70
230 IF F$="P" THEN SYSTEM "DUAL ON"
    : GOTO 70
240 IF F$="O" THEN SYSTEM "DUAL OFF"
    : GOTO 70
250 IF F$="E" THEN PRINT CHR$(26); "TC";
    CHR$(25); AS
    : END
260 IF F$="C" THEN CLS
    : GOTO 70
270 PRINT CHR$(26); "ERROR - UNKNOWN FUNCTION";
    CHR$(25)
    : GOTO 70
```

Read Screen

Alfred Pacheco
349 Westmoreland
Richland, Washington 99352

The Model II BASIC INTERPRETER is missing one command which most other versions of BASIC have. With the Model II BASIC you cannot read an existing character from the screen. You can, however, read a character that is about to be placed on the screen using the INKEY$ instruction. Unfortunately, if your program needs to know if a certain position on the screen is vacant or not, there is no command that enables you to do this.

The following machine language program was written for this very purpose. It can be entered using the DEBUG utility program and saved using the DUMP command and recalled later by your BASIC program.

The only variable needed by the subroutine is the position of the screen you wish to read. This position must be in Scroll Mode as described in the User Manual. If you are using the graphics mode, the coordinates must first be converted to Scroll mode before entering the subroutine. This can be done quite easily using the following function:

DEF FNVH (V,H) = 80*V + H
HV = FNVH(V,H)

where H is the horizontal position (0-79), and V is the vertical position (0-23). HV is then the desired position on the screen in the scroll mode. H,V and HV are in integer format.

USE OF THE SUBROUTINE IN A BASIC PROGRAM

When you enter BASIC, make sure that you leave the memory above EF00 reserved for the machine language program. Do this by typing in:

BASIC-M:61184

when entering BASIC.

Next, load and define the subroutine with the following:

SYSTEM"LOAD SCREEN/MAC"
DEFUSR0 = &HEF00

To use the subroutine the command

HV = USR0(HV)

is used. HV is the screen position you wish to read in scroll mode. HV is in integer format.

The Hex value of the character will be returned in HV. If there was no character there, then HV = 32. The Hex code can be converted to the actual character by command

A$ = CHR$(HV)

LOADING THE PROGRAM

To enter the machine language program, follow these steps:

1). When in TRSDOS READY, enter "DEBUG ON". After the response "DEBUG ON", enter "DEBUG".
2). When in DEBUG mode, enter "M" ⟨ENTER⟩ "EF00" ⟨ENTER⟩.
3). Press the ⟨F1⟩ key.
4). Enter the hex code as shown in Figure 1.
5). Press the ⟨ESC⟩ key.
6). Press the ⟨S⟩ key.
7). Type in "DUMP SCREEN/MAC START=EF00, END=EF24".

The program that follows (Figure 2) illustrates the use of this subroutine. The program will ask you to input the coordinates of the spot on the screen you wish to read and echo the character at the present cursor location.

TRS-80 MODEL II DEBUG PROGRAM

```
EFØØ    EB ED 53 23 EF 1A 6F 13    1A 67 ØE 5Ø Ø6 Ø1 3E 17
EF1Ø    CF 45 16 Ø1 2A 23 EF 3E    ØB CF 2A 23 EF 23 36 ØØ
EF2Ø    3E Ø2 C9 ØØ ØØ ØØ ØØ ØØ    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
EF3Ø    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
EF4Ø    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
EF5Ø    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
EF6Ø    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
EF7Ø    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ    ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```

Figure 1

```
1Ø CLS
2Ø PRINT "THIS IS A SAMPLE OF THE TEXT ON THE
    SCREEN"
3Ø PRINT "TO READ ANY LETTER IN THIS TEXT CHOSE
    THE PROPER"
4Ø PRINT "COORDINATE OF THAT LETTER USING THE
    GRAPHIC COORDINATE SYSTEM"
5Ø SYSTEM "LOAD SCREEN/MAC"
6Ø DEFINT H, V
7Ø DEFUSRØ=&HEFØØ
8Ø DEF FNVH(V, H)=8Ø*V+H
9Ø INPUT "H"; H
1ØØ INPUT "V"; V
    : PRINT "";
11Ø HV=FNVH(V, H)
12Ø HV=USRØ(HV)
13Ø PRINT CHR$(HV)
14Ø GOTO 9Ø
```

Figure 2

```
ØØ1ØØ          EX    DE,HL       ;PUT ASA ADDR.  IN DE REG
ØØ2ØØ          LD    (ADDR),DE   ;SAVE ASA ADDRESS IN ADDR
ØØ3ØØ          LD    A,(DE)
ØØ4ØØ          LD    L,A         ;GET ARGUMENT AND STORE
ØØ5ØØ          INC   DE          ;IT IN REG PAIR
ØØ6ØØ          LD    A,(DE)      ;HL
ØØ7ØØ          LD    H,A         ;
ØØ8ØØ          LD    C,8Ø        ;CONVERT SCROLL COOR. TO GRAPHIC COOR.
ØØ9ØØ          LD    B,1         ;BY DIVIDING ARGUMENT
Ø1ØØØ          LD    A,23        ;BY 8Ø
Ø11ØØ          RST   8
Ø12ØØ          LD    B,L         ;LOAD ROW POSITION INTO REB B. COLUMN
Ø13ØØ          LD    D,1         ;POSITION IN REG C. READ ONE CHARACTER
Ø14ØØ          LD    HL,(ADDR)   ;AND STORE IN ASA
Ø15ØØ          LD    A,11        ;
Ø16ØØ          RST   8
Ø17ØØ          LD    HL,(ADDR)   ;SET HIGH ORDER BYTE OF ASA
Ø18ØØ          INC   HL          ;TO Ø
Ø19ØØ          LD    (HL),Ø
Ø2ØØØ          LD    A,2
Ø21ØØ          RET               ;RETURN TO BASIC
Ø22ØØ  ADDR:   DW    Ø
Ø23ØØ          END
```

Figure 3

General Electric ASCII Sequential Files

Richard Halloran
155 Jackson Street No. 1704
San Francisco, CA 94111

TRSDOS for Models I, II and III supports two types of data files—Sequential and Random Access. There is a third type called General Electric ASCII sequential files or merely "terminal" files as they were used on devices similar to the Telex terminals.

Terminal files work on the TRS-80s and are very simple to set up and use. To illustrate, go into BASIC and type "AUTO" and then type in the following:

```
1Ø DICK,145,MYSELF
2Ø EDWARD MILES HALLORAN,2Ø2,MY NEPHEW
3Ø EDWARD,167,MY BROTHER
```

Now save that "program" to disk using the command:
SAVE "NAMES",A
which saves the "program" in ASCII form as a data file!

Now type NEW and enter the following program to access the "NAMES" data file.

```
1Ø CLEAR 5ØØ
    : CLS
2Ø OPEN"I",1,"NAMES"
3Ø FOR N=1 TO 3
4Ø INPUT #1,N$,W,R$
    : 'FIRST TIME THE FIELDS ARE NAMED !
6Ø TW=TW+W
7Ø PRINT MID$(N$,4) TAB(24);W TAB(3Ø) R$
8Ø NEXT
9Ø CLOSE
1ØØ PRINT
11Ø PRINT USING "THE TOTAL WEIGHT IS ### POUNDS"; TW
12Ø PRINT USING "THE AVERAGE WEIGHT IS ### POUNDS
    ";TW/3
```

Now, save this program to disk as "NAMES1" and then run it. The results should be as follows;

```
DICK                      145  MYSELF
EDWARD MILES HALLORAN     282  MY NEPHEW
EDWARD                    167  MY BROTHER

THE TOTAL WEIGHT IS 514 POUNDS
THE AVERAGE WEIGHT IS 171 POUNDS
```

Now let's see what is going on here. First we entered our data list consisting of three records with three fields each - a name, the persons weight, and the persons relationship to me. Then we saved the list in ASCII to disk and thus created a data file called "NAMES".

Next let us look at our BASIC program "NAMES1" which accesses and manipulates the data in "NAMES." Line 10 reserves string memory and clears the screen. Lines 30 and 80 set up a loop to input all the records. Line 40 accesses "NAMES" in sequential input mode and "declares" the fields N$,W, and R$. Line 60 accumulates the numeric variables fielded as W. Line 70 tabulates and prints the records. Note that the MID$(N$,4) prints all of the remainder of the field N$ after the third character. This suppresses the printing of the BASIC line numbers. Line 90 closes the file and lines 110 and 120 formats the printout of the results of the calculations. (Note that the numeric variable "W" does not have to be "converted" or "made" as in random access.)

What are some of the advantages of terminal files?

1. Very simple to learn how to use. Variable length fields
2. Data can be entered faster than if using input prompts.
3. The data file can be loaded into BASIC and edited.
 A. Line (Record) Editing
 B. Add, insert, delete records, add fields
 C. Scroll for checking data, etc.
4. Merge and append data files.
5. Insert EOF flags to make jump to successive drives.
6. No restrictions on the "driver" programs.
7. Efficient use of disk space

Model II Bugs, Errors, and Fixes

NOTE TO USERS:

The following program changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that the changes should be made, but you do not feel qualified to make the changes yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Services in Fort Worth for assistance.

CHANGES TO BASIC PROGRAMS

There are general procedures that need to be followed when any corrections are made.

1. Make a backup of the disk that contains the program to be corrected. Changes should be made on the backup copy.
2. Load the program to be changed by typing LOAD "filename" where filename is the name of the program to be modified.
3. Make the line changes indicated in the fix. For existing line numbers, edit or retype the line to match the one in the fix. New lines should be entered.
4. Save the corrected program (the one now in memory). Type SAVE"filename" ⟨**ENTER**⟩ where filename is the name of the program that has been modified.
5. Now make a backup of the corrected diskette.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHES to the backup copy of the diskette.
2. Apply PATCHES according to the information given in your TRSDOS manual.

RUNCOBOL

Users of the following programs:
26-4601 General Ledger
26-4604 Accounts Receivable
26-4605 Accounts Payable
26-4607 Order Entry/Inventory Control
26-4608 Sales Analysis
26-4703 COBOL Development System
26-4704 COBOL Runtime

All of the COBOL programs in the above list include the program RUNCOBOL. There are several sets of PATCHes that need to be made to the program RUNCOBOL. The first PATCH should only be applied to the programs specified in Correction 1.

The PATCHes in the other corrections should be applied to all the programs in the above list.

Correction 1—If one of the following programs is abnormally terminated, a problem may occur. An abnormal termination is one in which you suffer a power loss, press the ⟨**RESET**⟩ key, press the ⟨**BREAK**⟩ key, or turn your machine off while running the program. Conversely, a normal termination is one where the program is exited according to program instructions.

When an abnormal termination occurs, any data files you were working with at the time are rendered unuseable, even though they may appear to still be good. While some programs detect an abnormal termination as an ERROR 98, the following programs do not.

26-4601 General Ledger
26-4604 Accounts Receivable
26-4605 Accounts Payable
26-4607 Order Entry/Inventory Control
26-4608 Sales Analysis

You must—GO TO YOUR BACKUP. For Hard Disk users, RESTORE your system.

If you continue processing, the results will be unpredictable. Remember that the first thing you should do after bringing up backup data files as your new working data files is to create a new set of backup data files.

We have developed the patch below for RUNCOBOL. This patch will assist you in determining if you have a problem resulting from an abnormal termination. However, it is not the complete answer. When this patch has been applied and the program determines that the error has occurred, the following message will be displayed:

(DATA FILE NAME) ERROR 34

By pressing ⟨**ENTER**⟩ all files that are currently open will be closed, and the program will return to TRSDOS Ready. In this way you will know that you should go to your backup.

```
PATCH RUNCOBOL A=6E9Ø, F=98, C=34 14
```

NOTE: This PATCH should only be used with the programs indicated in Correction 1. It could cause inaccurate error detection if used with any other COBOL programs.

Correction 2—When attempting to add records to an indexed or relative file, COBOL improperly handles a disk full condition (error 24). The indices of indexed files are improperly changed. As a consequence, the next attempt to write a record produces an invalid index error (error 98).

At this point, the user can only revert to his most recent back-up copy, because the invalidated file cannot be recovered.

At TRSDOS READY, apply the following PATCHes to correct the above problem.

```
PATCH RUNCOBOL A=761D F=3Ø14FD7Ø71FD717Ø21 21
                       C=DØ2125ØØ19EB3E2CCF 2F
PATCH RUNCOBOL A=7626 F=25ØØ19EB3E2CCF28Ø4 14
                       C=2ØØ7FD7Ø71FD717ØC9 39
```

GENERAL LEDGER (26-4501)

The batch total does not equal the Document Total after document printing. This problem can also occur during posting. The problem is corrected by making the following changes to the program Txentry. (Version 1.2)

Line: `145Ø IFHI<LOTHEN14Ø5ELSEFORI=ØTONA-1 1`
`:S#(I)=Z#:NEXT:BT#=Z#`

Add the following line:

```
6345 IFIN#>ØTHENBT#=BT#+IN#
6740 IN#=BT#:LPRINTTAB(28)"BATCH TOTAL"TAB(42););
     :GOSUB51ØØ:LPRINT" ":L=L+1:RETURN
```

DELETE Line 6730
SAVE"Txentry" to store the program changes.
Make the following changes to the program Txpost.

Line:
```
42Ø FORJ=ØTONA-1:S#(J)=Z#:NEXT:BT#=Z#:RETURN
```

Add the following new lines:

```
74Ø IN#=BT#:LPRINTTAB(28)"BATCH TOTAL"
     TAB(42);:GOSUB51ØØ:LPRINT"  ";L=L+1:RETURN

3345 IFIN#>ØTHENBT#=BT#+IN#
```

DELETE LINE 730
SAVE"Txpost" to store the program changes.

PAYROLL (26-4503)

Following are corrections to separate problems with Payroll.

Correction 1—The Federal unemployment insurance rate factor has been reported to be to the third decimal place. The Payroll program will allow changes of up to the second decimal place. (Version 1.0)

Make the following changes to the program CHNGCO.

Line:
```
162Ø CLEAR2ØØØ:GOSUB4ØØ:CLS:PRINTCF$
     :PRINT@(2,16) ,R$"  O T H E R  R A T E S
     A N D  L I M I T S "N$:DIMFT#(9),FT(11)
     :F9$="###,###,##-": F8$="####":F7$=
     LEFT$(F9$,1Ø)+"#-"
```

Line:
```
167Ø FORI=1TO8:PRINT@(I+6,Ø),R$;USING" ## ";
     I;:PRINTN$;TAB(1Ø);C6$(I);:FORK=1TO2
     :J=VAL(MID$(C7$(I),2*K-1,2)):IFJ>ØANDI=5ANDK=1
     THENPRINTTAB(K*13+12);USINGF7$;FT#(J);
     ELSEIFJ>ØTHENPRINTTAB(K*13+12);USINGF9$;FT#(J);
```

Line:
```
176Ø PRINT@(19,Ø),EL$:PRINT@(19,8),P$;C6$(I);" ";
     :FL=8:GOSUB3ØØ:IFCF=1ORCF=2THENK=2:NEXT
     :GOTO171ØELSEIFLEN(IN$)>ØTHENFT#(J)=VAL(IN$)
     :IFI=5ANDK=1THENPRINT@(I+6,K*13+12),USINGF7$;
     FT#(J);ELSEPRINT@(I+6,K*13+12),USINGF9$;FT#(J);
```

SAVE"CHNGCO" to save the changes in the program.

Correction 2—Two new steps have been added to the calculation process for Minnesota State Tax Tables. The additions involve a 7% surcharge. The necessary program changes can be implemented by using the following procedure. (Version 2.0 & Prior)

Make the following changes to the program STATETAX.

To change Step 6 of the Minnesota Tax Calculation edit line 1820 to read as follows:

Line:
```
182Ø NEXT:PRINT#1,USINGL$;4800;:PRINT#1,
     "E#(I)=INT(E#(I)*1.Ø84#*1ØØ#+.5)/1ØØ#
     :IFE#(I)<ØTHENE#(I)=Ø:RETURNELSERETURN"
```

To change Step 2 of the Minnesota Tax Calculation add the following line.

Line:
```
1785 PRINT#1,USINGL$;SN+35;:PRINT#1,"G#=G#*.9Ø9DØ"
```

SAVE"STATETAX" to save the changes in the program.

Correction 3—In the Prepare Checks section of the Model II payroll program there is a problem. If you enter (choose to prepare a check for) an employee number that is greater than the total number of employees in your Payroll system the number entered will replace that of the employee currently displayed. (Version 2.0 and Prior)

Make the following changes to the program INPUT.

Line:
```
51Ø IFCF<>5THENIFCF<>ØTHEN5ØØELSEC=VAL(IN$)-EO
    :IFC<1ORC>LOF(3)THEN5ØØELSEN=C
```

Line:
```
52Ø IFN<1ORN>LOF(3)THEN5ØØELSEIFCF=ØTHENN1=
    INSTR(C1$,CHR$(N))
```

SAVE"INPUT" to store the program changes.

ACCOUNTS PAYABLE (26-4505)

Following are two sets of corrections for Accounts Payable.

Correction 1—Occasionally you may need to edit YTD purchases/payments. Make the following changes to the program APS/BAS. (Version 1.0)

Line:
```
2ØØ IFW=1THEN22ØELSEIFW$=CHR$(8)THENPRINTW$;FL$;B1$;
    :W=W-1:MID$(IN$,W,1)=FL$:GOTO14Ø
```

Line:
```
18ØØ GOSUB85ØØ:PRINT@(23,42),;:FL=1:SZ=1
     :GOSUB1ØØ:SZ=Ø:IFCF=1THEN1ØØØELSE
     IFCF=5THEN18ØØ
```

ADD the following new lines.

Line:
```
222 IFSZ<>1THEN23ØELSEIFASC(W$)=19THENPRINT@
    (22,2Ø),"YTD PURCHASES & CREDITS";:INPUTN#
    :MI#=A1#-N#:A1#=N#:N#=A1#:GOSUB5ØØØ:AA$=V$
    :GOTO226
```

Line:
```
224 IFASC(W$)=16THENPRINT@(22,2Ø),"YTD PAYMENTS &
    DEBITS";:INPUTN#:MI#=N#-A2#:A2#=N#:GOSUB5ØØØ
    :AB#=V$ELSE23Ø
```

Line:
```
226 GOSUB9Ø4Ø:PRINTN$;:GOSUB84ØØ:GOSUB92ØØ
    :IN$="X":RETURN
```

Line:
```
5ØØØ CP#=CP#-MI#:W#=ABS(N#)*1ØØ+.1:X=W#/D1#
     :W#=W#-X*D1#:V$=CHR$(X-(N#<Ø)*128)
     :X=W#/D2#:W#=W#-X*D2#:V$=V$+CHR$(X)
     :X=W#/D3#:W#=W#-X*D3#:V$=V$+CHR$(X)+
     CHR$(W#):RETURN
```

SAVE"APS/BAS" to save the changes in the program.

Run the program using the following options which are accessed from the VENDOR FILE maintenance menu:

CTRL S TO Edit PURCHASES
CTRL P TO Edit PAYMENTS

Correction 2—The check number does not increment if a check is voided (pre-printed checks only). Make the following changes to the program APCHECKS/BAS. (Version 2.0 and Prior)

Line:
```
2Ø6Ø IC=IC+1:IFIC=14THENLPRINTL1$:GOSUB8ØØØ
     :IC=Ø:CF=Ø:LC=LC+1
```

ADD this line:

Line:
```
3Ø75 IC=IC+1:IFIC=15THENLPRINT"":LPRINTL$
     :LPRINT"******* CHECK NUMBER "CN" VOIDED
      ******* - CONTINUED ON CHECK NUMBER
     "CN+1:LPRINTLL$:CN=CN+1:IC=Ø
```

SAVE"APCHECKS/BAS" to save the changes in the program.

MAILING LIST II (26-4507)

Once a given capacity is reached, the Mailing List program will not expand, and an error messages is displayed. (Version 2.0)

Make the following changes to the program SETUP/BAS.

Line:
```
127Ø IFRV<=ØORRV>CPTHEN125Ø
```

Add the following line:

Line:
```
985 IFM=85THENCP=3ØØØELSECP=2ØØØ
```

SAVE"SETUP/BAS" to save the changes in the program.

If you have already set up your Mailing List file then you need to complete the additional following steps.

At TRSDOS READY type in SETUP ⟨**ENTER**⟩.
Type ⟨**R**⟩ followed by the name of the system.
Press the ⟨**F2**⟩ key.
Press the ⟨**F1**⟩ key.

The system will show maximum capacity but will expand on the disk, one record at a time.

STATISTICAL ANALYSIS (26-4540)

Variable names do not change as variables change in the Correlation Matrix. (Version 1.0)

The problem is corrected by making the following changes to the program CORRL/BAS.

Line:
```
120 PRINT:LL%=MEM/8-1000:DIMX#(LL%),B$(300),
    V$(12),X$(12),VN$(12),NN$(12)
```

Line:
```
2060 ONERRORGOTO0:NR%=CVI(NR$):NV%=CVI(NV$)
     :FORJ=1TO12:VN$(J)=V$(J):NN$(J)=V$(J):NEXTJ
```

ADD the following new line:

Line:
```
2140 FOR DD=1 TO NV:VN$(DD)=NN$(WV%(DD)):NEXT DD
```

Type SAVE"CORRL/BAS" to save the changes in the program.

ACCOUNTS RECEIVABLE (26-4554)

If option 10 is chosen at the MAIN MENU so that system initialization is exited and you are returned back to the MAIN MENU, a BASIC Error Code 8 in line 51060 will occur. This is not a fatal error, and you may continue execution of the program by pressing <**ENTER**>. To prevent this error message from occurring make the following change to the program ARSI/BAS. (Version 1.1)

Line:
```
51060 IFR$<>"END"THEN RETURN ELSE GOTO 48000
```

Type SAVE"ARSI/BAS" to save the change.

RSCOBOL GENERATOR (26-4707)

Due to some misspelled words, COBOL Generator has a flaw in the code it produces. While this problem is not crucial, it does necessitate using an Editor to make corrections prior to compiling which brings about another problem.

The system is capable of generating 500K of code, but the CEDIT module only accommodates 32K of code thereby limiting the editing capability to 32K. Putting the system on Hard Disk does not increase the capacity of the Edit function. Our Text Editor (26-4710) does not work on Hard Disk. However, if you are using the Hard Disk in generating your code you can FCOPY it to 2.0a and then use the Text Editor (26-4710) to edit the code. Then FCOPY the file back to Hard Disk and compile it.

BISYNC (26-4715)

The BiSync package will lock up during Log On when interfacing with TSO running with TCAM. The following patch is to be used with version 3.1 of the BiSync package to correct the problem.

```
PATCH BIS3270 A=7203 B=FE27 C=FE26
```

Computer Clubs

THE COMPUGUILD
c/o Ben F. Davis
3513 Tripp
Amarillo, TX 79121
1-806-372-1898
1-806-358-2202

COMPUTER CLUB/TRS-80 USERS GROUP
c/o Harriette Gross
John C. Hart Memorial Library
1130 Main Street
Shrub Oak, NY 10588
1-914-245-5262

JACKSON AMATEUR COMPUTER SOCIETY (JACS)
c/o C.B.C.C. Inc.
2355 Camp Baker Rd.
Medford, OR 97501
1-503-535-6883

MICRO-80 GROUP TRS-80 USERS
c/o Bob Walters
P.O. Box 1472
Eugene, OR 97440-1472

NORTH IOWA TRS-80 USERS GROUP (NITUG)
c/o Chuck Prickett, Jr.
1705 South Coolidge
Mason City, IA 50401
1-515-423-3198

NORTHERN ILLINOIS COLOR COMPUTER CLUB
c/o John Bowman
9346 Landings Square
Des Plaines, IL 60016

RICHMOND TRS-80 USERS' GROUP
c/o MC Studio
4115 Hopkins Rd.
Richmond, VA 23234

SILICON VALLEY COMPUTER CLUB
GTE Systems Bldg. 3
Central Expressway & Whisman Rd.
Mountain View, CA 94043

SILVERLAKE DISTRICT COMPUTER CLUB
2475 Moreno Dr
Los Angeles, CA 90039
1-213-666-2116

SOUTH BAY TRS-80 USERS GROUP (SBUG)
Dysan Bldg.
5401 Patrick Henry Dr.
Santa Clara, CA 95050

TRS-80 MODEL I/III USERS CLUB
c/o Timothy I. Moder, Ph.D.
1932 McGee Avenue
Berkeley, CA 94703
1-415-841-8686

TRS-80 USERS GROUP OF NORTH UTAH
c/o Marcel Kinard
3106 East 125th North
Layton, UT 84041
1-801-544-8970

TRS-80 MODEL I USERS GROUP OF WEST LOS ANGELES
c/o Mike Miller
10210 Woodbine St. #1
Los Angeles, CA 90034
1-213-836-4103

Pocket Computer II Start-Up Instructions

There have been several complaints of defective PC-2s "out of the box".

The complaints are similar in nature and generally take the form of "I can't get it to do anything without producing an error" or "I can enter a program while in the program mode, but when I switch to the run mode it has forgotten the program."

Both of these complaints are traceable to a single source. Page 14 of the PC-2 Owner's Manual is not as explicit as it should be regarding the steps required after replacing the batteries or pressing ALL RESET. The manual should state:

After inserting the batteries and turning the unit over, your display should show:

"NEW0:CHECK?"

Respond by performing the following:

Press ⟨CLEAR⟩

Type NEW0 and press ⟨ENTER⟩

Press ⟨SHIFT MODE⟩

Type NEW and press ⟨ENTER⟩

These two operations clear the PC-2 main and reserve areas of memory.

It may be necessary for you to perform these operations whenever you change batteries again or press ALL RESET.

Homework

Brian Cutler
P.O. Box 124
Cairo, NY 12413

This little program is another one of my pocket Memo programs. In school, I have a very hectic schedule. So, I found myself frequently forgetting things. Well, with this program, that soon became a thing of the past. If I keep this program loaded into my computer at all times, I am able to come up with a list of what I have to do each day. The program below still has my course names included in it. This is so that people can see what it does, and they can then modify the headings to suit their own needs.

You can input ten lines of seven characters each for each heading. The last line (line 10) must be an "E." The menu at line 30 allows you to ADD information, PRINT OUT the information stored, or END the session.

```
1 REM HMWK
2 REM VERSION 1.2
3 REM 11/1/81
1Ø: A$(1Ø)="ENG 232"
    A$(11)="CSI 4Ø4"
    A$(12)="CSI 21Ø"
    A$(13)="PHI 21Ø"
2Ø A$(14)="PHY 1Ø4"
    A$(15)="WORK"
    A$(16)="MEMO"
3Ø INPUT "ADD(A),PRINT(P),END(E);A$
    : IF A$="P" GOTO 88
4Ø IF A$="A" GOTO 2ØØ
45 IF A$="E" GOTO 5ØØ
5Ø GOTO 3Ø
88 PRINT "****************"
89 PRINT "* HOMEWORK AND *"
    : PRINT "*      MEMOS     *"
    : PRINT "****************"
    : PRINT " "
9Ø D=9
91 LET B=1Ø
92 D=D+1
    : IF D=17 GOTO 3Ø
93 PRINT"----------------"
    : C=B+1Ø
94 PRINT" ( ) ";A$(D)
95 PRINT " "; A$(C)
96 C=C+1
97 IF A$(C)="E" LET B=B+1Ø
    : GOTO 92
98 GOTO 95
2ØØ INPUT "DATE:";F$
    : D=9
2Ø1 LET B=1Ø
2Ø2 D=D+1
    IF D=17 GOTO 3Ø
2Ø3 C=B+1Ø
2Ø4 PAUSE A$(D)
2Ø5 INPUT "HMWK:";A$(C)
2Ø6 IF A$(C)="E" LET B=B+1Ø
    GOTO 2Ø2
2Ø7 C=C+1
2Ø8 GOTO 2Ø5
5ØØ PRINT " "
    PRINT " "
    PRINT "DATE:";F$
5Ø1 PRINT " "
    : PRINT " "
    : PRINT " "
    : BEEP 1Ø
    : END
```

Partial Sample Printout

```
****************
* HOMEWORK AND *
*     MEMOS    *
****************

----------------
( ) ENG 232
      READ
      SHORT
      STORY
      END
```

Use of the TRS-80 PC in Celestial Navigation

Donald Parson
4475 N. Ocean Blvd., #43-C
Delray Beach, FL 33444

PROGRAM ONE

For those interested in celestial navigation, here is a program for the PC to solve the spherical triangle and come up with intercept and azimuth without the use of tables. All that is required is sextant, watch and almanac.

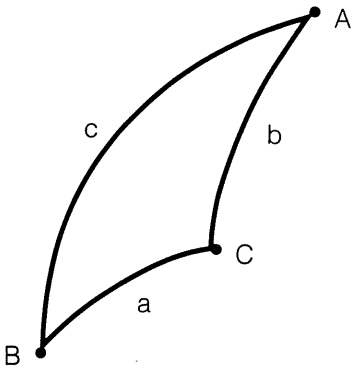


Figure 1

The spherical triangle (Fig. 1) may be solved as follows:

COS a = COS b * COS c + SIN b * SIN c * COS A. (1)

Solving for "a" will yield "Hc", the computed altitude.

COS C = (COS c - COS a * COS b)/(SIN a * SIN b) (2)

Solving for "C" will yield "Z", the azimuth angle.

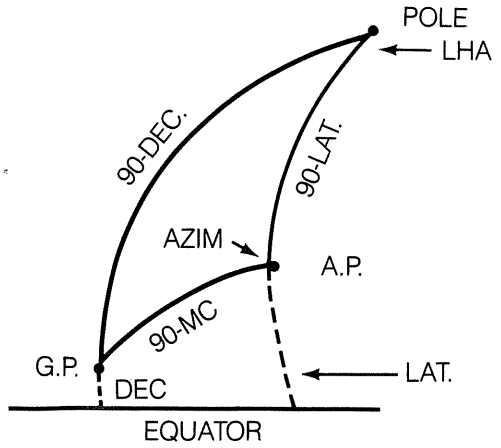


Figure 2

Fig 2. shows how the various inputs are worked into the triangle. Fig. 3 shows what happens when the declination and latitude are of 'contrary name'. "A.P." is the assumed position of the observed body, as taken from the almanac.

This program will compute LHA directly from GHA and longitude, thus disposing of another chore and another chance of human error.

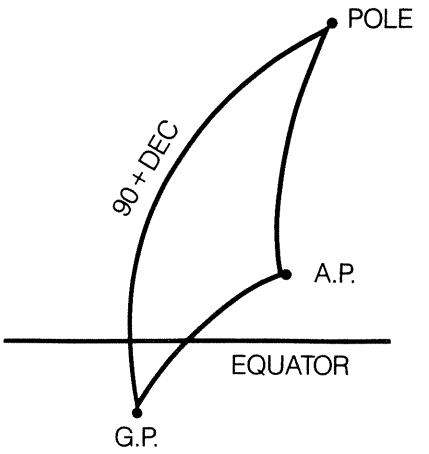


Figure 3

```
1Ø CLEAR
2Ø INPUT "LAT,DEG? "; A
21 INPUT "    MIN? "; X
22 A=A+X/6Ø
   : X=Ø
25 INPUT "IF NORTH LAT, DIAL 1   "; W
3Ø INPUT "DEC,DEG?   "; B
31 INPUT "    MIN?   "; X
32 B=B+X/6Ø
   :  X=Ø
33 INPUT "IF SAME NAME, DIAL 1   "; X
34 IF X<>1 THEN 55Ø
35 INPUT "GHA,DEG?  "; O
36 INPUT "    MIN?  "; P
37 O=O+P/6Ø
38 INPUT"LONG,DEG?  "; P
39 INPUT"     MIN?  "; U
4Ø P=P+U/6Ø
41 INPUT "IF WEST,DIAL 1   "; U
42 IF U<>1 LET P=-P
43 IF O<P LET O=O+36Ø
44 C=O-P
45 INPUT "HO,DEG?   "; N
46 INPUT "   MIN?   "; Y
47 Q=6Ø*(Y-INT Y)
48 N=N+(INT Y/1ØØ)+Q*.ØØØ1
5Ø D=9Ø-A
   : E=9Ø-B
   : DEGREE
6Ø F=COS D*COS E
   : H=COS C
7Ø G=SIN D*SIN E
8Ø I=ACS(F+G*H)
   : J=DMS(9Ø-I)
81 Q=1ØØ*J-INT(1ØØ*J)
82 R=INT(1ØØ*J)+Q/.6
83 S=INT(R/1ØØ)
84 T=1ØØ*(R/1ØØ-S)
   : BEEP 2
9Ø PAUSE "HC="; S; "DEG "; USING "####.##"; T; "MIN"
91 IF J<N THEN 97
92 V=6Ø*(DEG J-DEG N)
96 PRINT V; "    AWAY"
   : GOTO 1Ø5
97 V=6Ø*ABS(DEG J -DEG N)
1ØØ PRINT V; "   TOWARD"
1Ø5 K=COS E-(COS I*COS D)
11Ø L=SIN I*SIN D
```

```
120 M=ACS(K/L)
130 PAUSE " Z= "; M
    : GOTO 210
140 IF C<180 THEN Z=360-M
210 IF W=1 THEN 300
220 IF C>180 THEN 500
230 Z=180+M
250 PRINT "ZN= "; Z
    : GOTO 10
300 IF C>180 THEN 400
310 Z=360-M
    : GOTO 250
400 Z=M
    : GOTO 250
500 Z=180-M
    : GOTO 250
550 B=-B
    : X=0
    : GOTO 35
600 END
```

PROGRAM TWO

The first program was for sight reduction only. This new one includes Great Circle Distance and initial azimuth (course) from present position to destination. It also furnishes the numbers required for the use of a star finder (HO 2102-D).

Line 10 starts the Great Circle program, line 690 the Star Finder and 905 the Sight Reduction. You will note that the two navigational programs share some of the operations, inasmuch as they use the same basic formulae.

COMMENTS ON "F", SIGHT REDUCTION:

Note that this program can yield an intercept from the DR position, instead of from an 'assumed position' as required when using the tabular solution. This is a big benefit. In line 916 the Z is held for the later determination of azimuth from azimuth angle C in the triangle (see Fig. 4). This determination occurs in line 310, via a "GOTO". Line 926 provides the proper sign for longitude when it is (eventually) used in producing LHA (Angle A in the triangle). Line 940 deals with the situation when the celestial body is on the other side of the equator from the observer. Line 210 converts latitude and declination to b and c in the triangle. In line 262, V would have been zero, had we been using the Great Circle program.

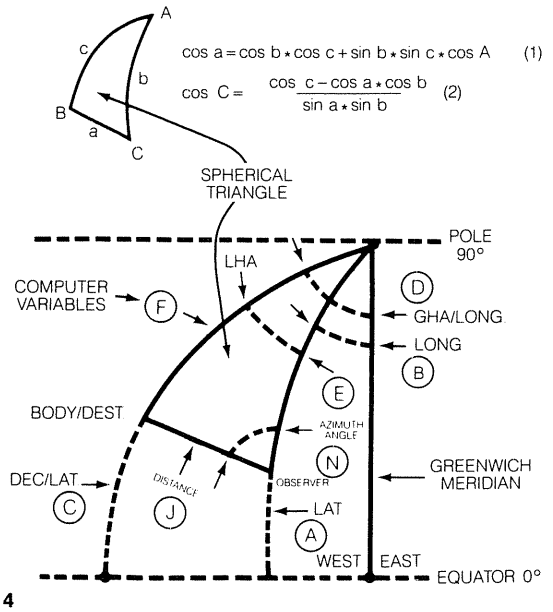


Figure 4

Thus we can adapt the use of the triangle solution according to our requirements. V being 1, we enter the H.O. data and subtract it from (compare it to) the triangle solution. This yields the Intercept, either 'toward' or 'away'. We then jump to line 280 for the azimuth answer, which would be common to the two programs.

COMMENTS ON "D", GREAT CIRCLE:

Line 10 "D" merely changes the wording for some of the inputs, substituting the destination for the celestial body. Line 200 corrects the situation where, in the celestial solution, GHA is always measured westward, whereas in the Great Circle solution the longitude can go either way. V is made to be zero, so that later on (line 270) we print the proper things. In line 260 we multiply the angle by 60 to get miles.

COMMENTS ON "S", STAR FINDER:

"S" of course could be omitted, but I have found that it reduces the star finder set up time from, say, 20 minutes to about 5. It also eliminates many chances for mistakes.

Lines 720 and 730 set the longitude of the observer. They could have been written:

```
700: INPUT "LONGITUDE,DEGREES"; A
710: INPUT "          MUNUTES"; B
720: A=A+B/60
730: INPUT "IF WEST,DIAL 1"; B
```

I happen to be at 80 degrees 3.5′ W, and expect to remain for a while.

I hope that the foregoing is of interest. It certainly was interesting to compose it. I wound up with only 13 steps and 1 memory left!

```
5 PRINT " SHFT
    :  S,F OR D"
10 "D"
    : CLEAR
20 INPUT "DEP LAT DEG  "; A
30 INPUT "        MIN  "; B
40 A=A+B/60
    :  B=0
50 INPUT "N,DIAL 1  "; Z
60 INPUT "LONG DEG  "; B
70 INPUT "     MIN  "; C
80 B=B+C/60
90 INPUT "W,DIAL 1  "; Y
100 IF Y=0 LET B=-B
110 INPUT "ARR LAT DEG  "; C
120 INPUT "        MIN  "; D
130 C=C+D/60
140 INPUT "SAME NAME,DIAL 1 "; X
150 IF X=0 LET C=-C
160 INPUT "LONG DEG  "; D
170 INPUT "     MIN  "; E
180 D=D+E/60
190 INPUT " W DIAL 1  "; W
200 IF W=0 LET D=-D+360
    :  V=0
210 E=90-A
    :  F=90-C
    :  DEGREE
220 G=COS E*COS F
230 H=D-B
240 IF H<0 LET H=H+360
250 I=SIN D*SIN F*COS H
260 J=ACS (G+I)
    :  K=60*J
262 IF V=1 THEN 400
270 PRINT "GRT CIR DIST=";USING "#######.#"; K; MI"
280 L=COS F-(COS J*COS E)
```

```
29Ø M=SIN J*SIN E
3ØØ N=ACS (L/M)
31Ø IF Z=1 THEN 35Ø
32Ø IF H>18Ø THEN 38Ø
33Ø O=18Ø+N
34Ø PRINT "AZIMUTH=";USING "####.#"; O
341 END
35Ø IF H>18Ø THEN 37Ø
36Ø O=36Ø-N
   :  GOTO 34Ø
37Ø O=N
   :  GOTO 34Ø
38Ø O=18Ø-N
   : GOTO 34Ø
4ØØ INPUT " H.O.DEG.  "; P
41Ø INPUT "     MIN.  "; Q
42Ø P=P+Q/6Ø
   : Q=Ø
43Ø R=9Ø-J
   : Q=6Ø*ABS (R-P)
44Ø IF R<P THEN 46Ø
45Ø PRINT USING "####.#"; Q; "A"
   : GOTO 28Ø
46Ø PRINT USING "####.#"; Q; "T"
   : GOTO 28Ø
69Ø "S" CLEAR
72Ø A=8Ø+3.5/6Ø
73Ø B=1
74Ø IF B=Ø LET A=-A
75Ø INPUT " GHA ARIES,DEG  "; C
76Ø INPUT "          MIN  "; D
77Ø C=C+D/6Ø
78Ø D=C-A
79Ø IF D>36Ø LET D=D-36Ø
8ØØ INPUT " GHA,BODY,DEG  "; E
81Ø INPUT "          MIN  "; F
82Ø E=E+F/6Ø
83Ø F=C-E
835 IF E>C LET F=F+36Ø
84Ø PRINT " LHA,ARIES="; USING "####.#"; D
85Ø PRINT "R.A.BODY=";F
86Ø INPUT " GHA,BODY,DEG "; E
87Ø INPUT "          MIN "; F
88Ø E=E+F/6Ø
89Ø F=C-E
895 IF E>C LET F=F+36Ø
9ØØ PRINT " R.A. BODY=";F
   : GOTO 86Ø
9Ø5 "F" CLEAR
91Ø INPUT " DR LAT DEG "; A
912 INPUT "        MIN "; B
914 A=A+B/6Ø
916 INPUT "  N DIAL 1  "; Z
918 INPUT " DR LONG DEG  "; B
92Ø INPUT "          MIN  "; C
922 B=B+C/6Ø
924 INPUT "  W DIAL 1  "; Y
926 IF Y=Ø LET B=-B
928 INPUT "  GHA DEG  "; D
93Ø INPUT "      MIN  "; E
932 D=D+E/6Ø
934 INPUT "  DEC DEG  "; C
936 INPUT "      MIN  "; E
938 C=C+E/6Ø
94Ø INPUT " SAME NAME DIAL 1  "; X
942 IF X=Ø LET C=-C
944 V=1
   : GOTO 21Ø
```

PROGRAM THREE

For you navigators, here is a reserve program which will give you the coordinates of Latitude and Longitude directly from the altitudes of Polaris and any other body.

This we can accomplish with only a sextant and the almanac; no reduction tables required.

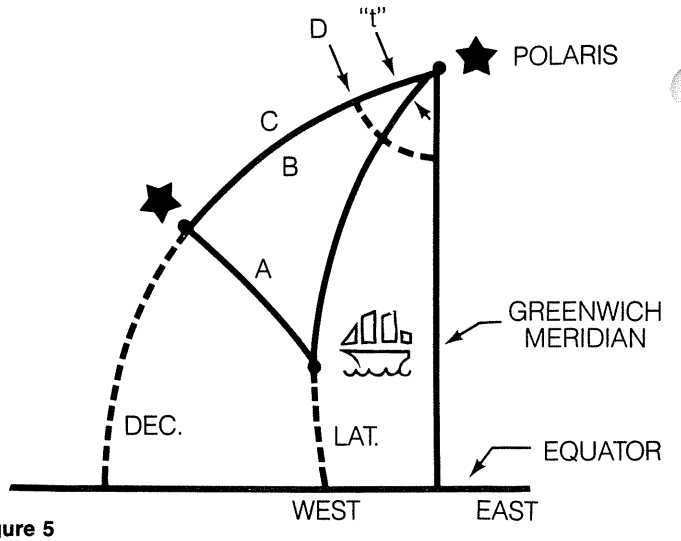


Figure 5

Shown in figure 5 is a spherical triangle, which is involved in our problem. What we do is to solve for the angle "t" in the relationship:

$$\cos\text{"t"} = \frac{\cos A - \cos B * \cos C}{\sin B * \sin C}$$

Our reserve program is written:

SHFT A: ACS((COS A - COS B * COS C)/(SIN B * SIN C))

The inputs are as follows:
A = 90 - Altitude of Body
B = 90 - Altitude of Polaris
C = 90 - Declination
D = Greenwich Hour Angle

You must also know whether you are East or West of Greenwich, whether the body bears Easterly or Westerly and that you are in the Northern Hemisphere. If the Declination is of opposite name, then:

C = 90 + Declination

To operate the program, the keying would be like this:

1. In West Longitudes, with the Bearing Easterly:

 ⟨D⟩ ⟨+⟩ ⟨SHFT⟩⟨A⟩ ⟨ENTER⟩

2. In West Longitudes, with the Bearing Westerly:

 ⟨D⟩ ⟨-⟩ ⟨SHFT⟩⟨A⟩ ⟨ENTER⟩

3. In East Longitudes, with the Bearing Easterly:

 ⟨3⟩⟨6⟩⟨0⟩ ⟨-⟩ ⟨D⟩ ⟨-⟩ ⟨SHFT⟩⟨A⟩ ⟨ENTER⟩

4. In East Longitudes, with the Bearing Westerly:

 ⟨3⟩⟨6⟩⟨0⟩ ⟨-⟩ ⟨D⟩ ⟨+⟩ ⟨SHFT⟩⟨A⟩ ⟨ENTER⟩

In each case, you will read the Longitude ater the last **⟨ENTER⟩**. If the result exceeds 360, you simply subtract 360.

The Latitude, of course, is equal to the Altitude of Polaris.

Transferring Data From Long Records Into Shorter Records

Profile
The small Computer Company

Written for the users of PROFILE II, PROFILE + and PROFILE III +

PROFILE Editor
P.O. Box 2910
Fort Worth, TX 76113-2910

Mr. Robert D. Lynch of Glassboro, New Jersey, writes: "I have a TRS-80 Model II with PROFILE II. Before I understood your patches described in the June 1981 'News' which allows me to shorten segments and increase my number of records, I already had a sizeable file which is now limited to approximately 1800 records. With the patches and shortened segments I have been able to create a new file which will open to approximately 3000 records. However, I have not been able to successfully transfer the records from my old file to the new file. How do I do it? In my old file I have used the second and third segments but have limited the totals of the field lengths to 128 bytes."

What is needed is a short BASIC program that will write the old 256 byte records into 128 byte records. This entails taking the first 128 bytes from the old records and writing them into 128-byte records in a new file.

Reading PROFILE files from BASIC was described in the November 1981 column. Using that information as a jumping-off point, you should be able to use this sample program to write the old to the new file:

At TRSDOS READY, open 2 files in BASIC: BASIC -F:2
Type in the following program:

```
10Ø  OPEN "D",1, "FILENAMENEXT
     : n "
11Ø  OPEN "D",2, "FILENAMENEXT
     : n ",128
12Ø  FIELD 1,128 AS IN$
13Ø  FIELD 2,128 AS OT$
14Ø  FOR R=1 TO LOF(1)
15Ø  GET 1,R
16Ø  LSET OT$=IN$
17Ø  PUT 2,R
18Ø  NEXT R
```

In place of "FILENAME" use the name of your file; if the name is less than 8 characters, pad it with zeroes—i.e., "SHORT000/EXT."

PROFILE data file extensions ("EXT") are /DAT for segment 2; /DA2 for segment 3; and /DA3 for segment 4. To move segments 2 and 3, Mr. Lynch will have to type in the program twice, using FILENAME/DAT the first time, and FILENAME/DA2 the second time.

Note: You cannot move the KEY—the first segment—with this program because key segments are written in blocks of 3 (1 from each of 3 records), and un-blocking and re-blocking the segments is quite another process. However, in the above example, the old key segment can be used for the new file, too.

The "n" at the end of the extension is the drive number. If you have only one drive, or want to write the old file into the new file on the same drive, your new file name must be different from the old file name. After the records are written into the new file, change the new name to the old name using the TRSDOS "RENAME" command.

Once you've verified that the records have been written successfully into the new file, you can kill the old file. If the old and new files have the same name, we suggest you take the diskette(s) holding the new file out of the system first, to avoid killing it by mistake.

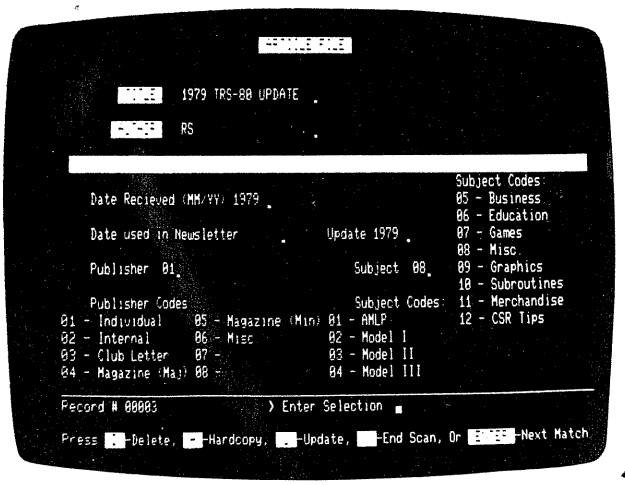
One question we often hear from PROFILE II and PROFILE PLUS users is, "What **are** all those programs on my diskette—what do they do?"

To answer this question once and for all—until we write more add-on programs, that is—we've come up with the following table.

Included are PROFILE II, PROFILE PLUS. Where you see "FILENAME," substitute your own file's name.

Program	**Menu Choice & Function**
PROFILE II	
M	PROFILE II menu
CREATION PROGRAMS	
CREATE/EFC	Menu choice 1—define files; creates FILENAME/MAP, KEY, DAT, DA2, DA3
CREATEX/EFC	Menu choice 2—define screens; creates FILENAME/PMP, PM1-5
LPFORM/EFC	Menu choice 3—define report formats; creates FILENAME/PRT, PR2-5

LBFORM/EFC	PROFILE II menu choice 4—define label formats; creates FILENAME/LB1-5
FIELDER/EFC	PROFILE II menu choice 5—define selections for SCRIPSIT; creates FILENAME/SL1-5 (SL = Selection List)
RUNTIME PROGRAM	
EXPAND/EFC	PROFILE II menu choice 6—expand files FILENAME/KEY, DAT, DA2, DA3; in single-segment files only, creates FILENAME/KX1-3.
CLERK/EFC	PROFILE II menu choice 7, Inquire, Update, Add; uses the indexes; uses screen formats FILENAME/PMP, PM2-5
PRINT/EFC	PROFILE II menu choice 8, print reports; reads in FILENAME/PRT, PR2-5
LABEL/EFC	PROFILE II menu choice 9, print labels; uses FILENAME/LB1-5
SELECTOR/EFC	PROFILE II menu choice 10, run record selection for SCRIPSIT; reads in FILENAME/SL1-5 and creates FILENAME/SR1-5 (SR = Select Records)
PROFILE PLUS	
MENUPLUS/EFC	The PROFILE PLUS menu
MAKEMENU/EFC	PROFILE PLUS menu choice 1—create a user's menu
K/EFC	A key file; it allows you to pass parameters to programs in a user's menu
CREATEM/EFC	PROFILE PLUS menu choice 2—define math formulas; creates FILENAME/MTH
INDEX/EFC	PROFILE PLUS menu choice 3—build index; creates FILENAME/IX1



New Journal Calls For Papers

Gerald H. Block
A LEARNING PLACE
6472 Moraga Avenue
Oakland, CA 94611

A new quarterly, THE JOURNAL OF COMPUTERS READING & LANGUAGE ARTS (CRLA), is ready to receive papers. The journal's purpose is to support the rapidly growing interest in computers and their relationship to reading, language arts and related issues. The theme of the journal will be pragmatic in perspective. It will emphasize presenting papers which have clear classroom/teaching implications.

The journal will be interdisciplinary and directed toward an audience of reading-language arts teachers, educational specialists, classroom teachers, educators of teachers and educational researchers.

Papers or requests for information should be sent to Gerald H. Block, CRLA, P.O. Box 13039, Oakland, CA 94661.

Radio Shack Agricultural Software Sourcebook

Radio Shack invites you to list your agricultural computer software in our new Agricultural Software Sourcebook.

Listing your software in the TRS-80 Agricultural Software Sourcebook could be the most effective advertising expenditure you will make in 1982: a listing will introduce a description of your programs to the owners of thousands of TRS-80 computers in the agricultural market, through the more than 6,000 Radio Shack Stores which will sell the sourcebook. No one else can make you an offer like this one.

To obtain a single listing in the forthcoming Radio Shack Agricultural Software Sourcebook (26-2774) for a period of one year, complete the registration form, submit a $10.00 check or money order to "Radio Shack", and mail them to the following address:

RADIO SHACK AGRICULTURAL SOFTWARE SOURCEBOOK
DEPARTMENT AX10—ONE TANDY CENTER
FORT WORTH, TEXAS, 76102

Registration forms can be obtained by writing department AX10 at this address. If you wish to list ten or more of your agricultural programs in the Radio Shack Agricultural Software Sourcebook, you can do so for only $5.00 per listing. This is a savings of 50% per program listing off the standard fee of $10.00 per listing.

To accept our special listing offer, just send 10 or more separate forms together with a check or money order for all the listings at the $5.00 per listing rate.

We invite you to join us by listing your agricultural software. The deadline for the first edition is November 30, 1982.

Planetarium

J.W. Hurley
3403 Onyx Street
Torrance, CA 90503

Have you ever wondered what the stars looked like but did not want to go out in the rain? Have you ever wondered which planet is visible and where? Have you ever wondered where Sirus is located? Have you ever wondered why I do not get on with it?

Here are some of the keys to the program.

Line	20	Insert your own value for LA based on your latitude. This runs from about 30 in the Southern United States to around 50 in the North. Not critical.
Line	50	Input either a carriage return or a three letter abbreviation of a constellation. Example: enter "SCO" and all the stars of Scorpius will be shown oversize.
Line	60	As above except the individual stars named will be accented. If "SUN" is entered part two will draw the sun and accent the planets.
Line	80	Prints instructions of what to do after the chart is complete.
Line	160	Converts date into days into the year. Lines 270-190 chew up the rest of the data so the next read starts with the star data.
Lines	220-790	Four part data statements, first the constellation name, then the star name if available, or a "Z" for a null, then the right ascension in hours and finally the north-south elevation. This information is taken from reference books and more stars can be included. We have named the 21 first magnitude stars and included most second magnitude start as unnamed ones.
Line	790	When data "XXX" is read the program knows the last information has been read.
Lines	820-840	Draws a cross that is directly overhead and a circle to indicate the horizon and labels East and West.
Lines	850-950	Reads data, plots position based on an arc around the North Star. PSETs a point. "T" is a factor that increases star time four minutes each day into the year as compared to solar time which we use.
Lines	970-980	Accent stars as requested at the start of the program.
Line	990	Says go back and read another set of data if "XXX" has not been read.
Line	1000	Says go do the solar system and print the data.
Line	1020	After all the above has been done the computer waits for your instruction. There is no prompt.
Line	1010	Just in case you wish to redraw at a later date or hour the program chews up the first data set so it can reread the star data.
Lines	1110-1250	Prints the time and sounds a signal that all is done as requested.
Lines	1260-1290	Plots how far above or below the celestial equator the sun is for the time of year. This factor, "E," is also used to locate the planets. They are not all right on the line of the sun but are very close. If all is right the sun will be all the way North about Jun. 22 and South for winter December 22.
Line	1290	Locates the sun by the hour and then gores sub to plot x-y coordinates again based on an arc as the stars were.

The balance of the program reads data based on the number of months into the time from 1-1-82. If a date outside the 82-85 requested this part is skipped.

Data is read for the middle of each month. Only in a few cases is this far off. Note that the data for VA (Venus) is given in degrees from the sun while the slower planets use the same right ascension as the stars used earlier.

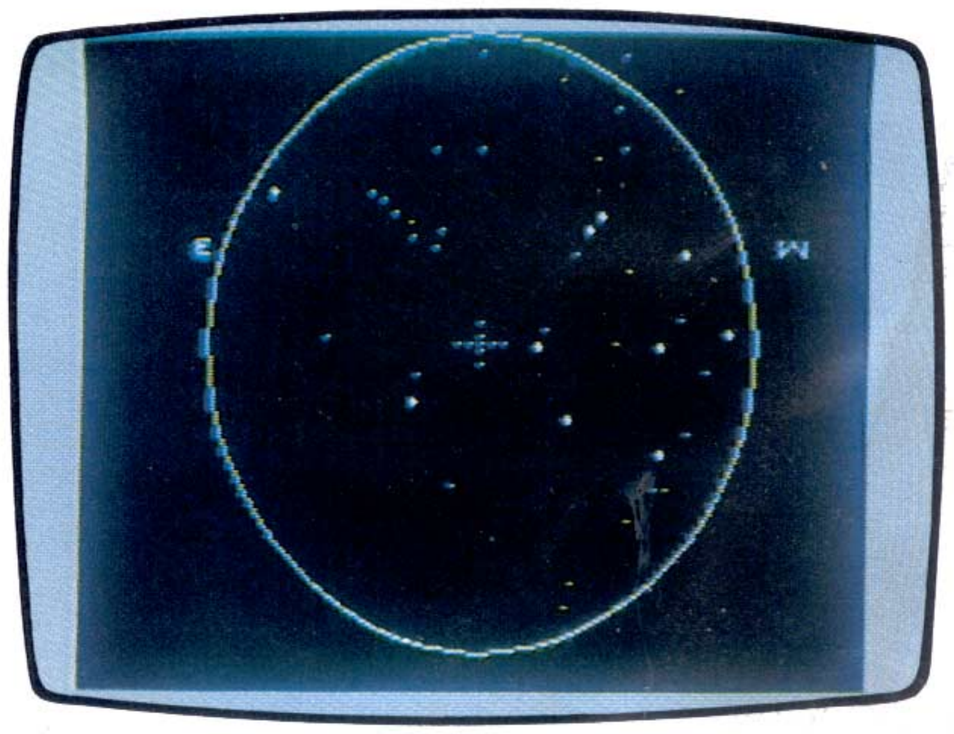
I hope other readers enjoy improving the program and adding to it as much as I have enjoyed developing it almost line by line. I see quite a few areas for improvement but you have to stop somewhere. It shows that the CC is good for many things beside making pretty patterns.

I have had my setup about a year and have taught it many things from playing Blackjack to figuring income tax. I am retired and dollars are scarce but this computer has been well worth the money in enjoyment received. I hope you will print many different kinds of programs to spark interest.

```
1Ø 'ENTER VIEWING LATITUDE FOR LA
2Ø LA=33
3Ø CLS
4Ø PRINT
     : PRINT
     : PRINT "       STAR MAPS"
     : PRINT
     : PRINT
```

```
5Ø PRINT "TO ACCENT ANY CONSTELLATION      TYPE
     IT'S FIRST THREE LETTERS"
     : INPUT E$
6Ø PRINT "TO ACCENT A STAR-TYPE IT'S FIRST THREE
     LETTERS. TYPE <SUN> TO   ACCENT SOLAR
     SYSTEM."
     : INPUT C$
7Ø C$=RIGHT$(C$, 3)
8Ø PRINT "AFTER CHART IS DISPLAYED HIT   SPACE
     BAR TO ADVANCE ONE HOUR  <E> TO DRAW
     CELESTIAL EQUATOR   <D> TO ENTER NEW DATE
     <Q> TO QUIT
9Ø INPUT "DATE(MM,DD,YY (82-85))"; M, D, YY
1ØØ IF M<Ø OR M>12 THEN 9Ø
11Ø IF D<Ø OR D>31 THEN 9Ø
12Ø INPUT "CHART TIME-USE STANDARD TIME    AND 24
     HOUR FORMAT TO NEAREST   HOUR."; H
13Ø H=INT(H)
     : DY=Ø
14Ø IF H<Ø OR H>24 THEN 12Ø
15Ø DATA Ø, 31, 28, 31, 3Ø, 31, 3Ø, 31, 31, 3Ø,
     31, 3Ø, 31, -99
16Ø FOR J=1 TO M
     : READ DD
     : DY=DY+DD
     : NEXT
17Ø READ Z
18Ø IF Z=-99 THEN GOTO 2ØØ
19Ø GOTO 17Ø
2ØØ DY=DY+D
21Ø PMODE 4, 1
     : PCLS
     : SCREEN 1, 1
22Ø DATA "ERI", "ACH", 1.6, -57
23Ø DATA "UMA", "Z", 1Ø.8, 61.8
24Ø DATA "UMA", "Z", 1Ø.8, 57
25Ø DATA "UMA", "Z", 11.7, 54
26Ø DATA "UMA", "Z", 12.1, 57
27Ø DATA "CAS", "Z", .6, 56.4
28Ø DATA "CEP", "Z", 21, 62.5
29Ø DATA "GRU", "Z", 22.6, -47
3ØØ DATA "AUR", "Z", 5.9, 45
31Ø DATA "CMA", "Z", 6.8, -26.8
32Ø DATA "CRU", "ACR", 12.4, -63
33Ø DATA "UMA", "Z", 12.7, 55
34Ø DATA "UMA", "Z", 13.3, 53
35Ø DATA "UMA", "Z", 13.6, 5Ø
36Ø DATA "TAU", "ALD", 4.6, 16
37Ø DATA "AQU", "ALT", 19.8, 9
38Ø DATA "SCO", "ANT", 16.5, -26
39Ø DATA "BOO", "ARC", 14.2, 19
4ØØ DATA "CRU", "BEC", 12.7, -6Ø
41Ø DATA "ORI", "BET", 5.9, 7
42Ø DATA "CAR", "CAN", 6.4, -63
43Ø DATA "AUG", "CAP", 5.2, 46
44Ø DATA "CYG", "DEN", 28.7, 45
45Ø DATA "FAU", "FOM", 22.9, -3Ø
46Ø DATA "CNE", "HAD", 14, -6Ø
47Ø DATA "GEM", "POL", 7.7, 28
48Ø DATA "CMI", "PRO", 7.6, 5
49Ø DATA "LEO", "REG", 1Ø.1, 12
5ØØ DATA "ORI", "RIG", 5.2, -8
51Ø DATA "CEN", "RIG", 14.6, -61
52Ø DATA "CMA", "STR", 6.7, -17
53Ø DATA "VIR", "SPI", 13.4, -11
54Ø DATA "LYR", "VEG", 18.6, 39
55Ø DATA "PEG", "Z", .1, 29
56Ø DATA "PEG", "Z", .1, 15
57Ø DATA "PEG", "Z", 23, 29
58Ø DATA "PEG", "Z", 23, 15
59Ø DATA "UMI", "Z", 1, 9Ø
6ØØ DATA "UMI", "Z", 14.7, 74
61Ø DATA "PHO", "Z", 4, -42, "CET", "Z", .4, -18,
     "AND",  "Z", 1, 35, "AND", "Z", 2, 42,
     "ARE", "Z", 2, 23, "CET", "Z", 2.3, -3
62Ø DATA "ORI", "Z", 5.7, -9.6, "AUR", "Z", 5.9,
     45,  "CMA", "Z", 6.3, -18, "GEM", "Z", 6.4,
     16.4, "CMA", "Z", 6.9, -26, "CMA", "Z", 7.3,
     -29
63Ø DATA "PER", "Z", 3, 41, "PER", "Z", 3.3, 5Ø,
     "ORI",  "Z", 5.4, 6.3, "TAU", "Z", 5.4, 28.6
64Ø DATA "GEM", "Z", 7.4, 32, "PUP", "Z", 8, -4Ø,
     "VEL",  "Z", 8.1, -47
65Ø DATA "HYD", "Z", 9.3, -8.6, "LEO", "Z", 1Ø.1,
     2Ø,  "LEO", "Z", 11.8, 14.7, "CRU", "Z",
     12.4, -57, "CEN", "Z", 12.7, -49, "CEN",
     "Z", 13.4, -53
66Ø DATA "CEN", "Z", 14.1, -36.2, "CEN", "Z",
     14.5,  -42, "LUP", "Z", 14.6, -47, "BOO",
     "Z", 14.6, 27
67Ø DATA "CBO", "Z", 15.5, 26.3, "SCO", "Z",
     15.9,  -22.5, "SCO", "Z", 16.7, -34.2,
     "OPH", "Z", 17.1, -15.7, "SCO", "Z", 17.5,
     -37, "OPH", "Z", 17.5, 12.5
68Ø DATA "SCO", "Z", 17.5, -43, "SCO", "Z", 17.6,
     -39,  "DRA", "Z", 17.7, 51.5, "SAG", "Z",
     18.3, 67, "SAG", "Z", 19.2, -26
69Ø DATA "CYG", "Z", 2Ø, 4Ø, "PAY", "Z", 2Ø.4,
     -56,  "CYG", "Z", 2Ø.6, 33.9, "PEG", "Z",
     21.6, 9.7, "GRU", "Z", 22.3, -47
7ØØ DATA "PER", "Z", 3.3, 5Ø
71Ø DATA "CAS", "Z", .1, 58
72Ø DATA "CAS", "Z", .5, 55
73Ø DATA "CAS", "Z", 1, 6Ø
74Ø DATA "CAS", "Z", 1.4, 58
75Ø DATA "CAS", "Z", 1.9, 62
76Ø DATA "ORI", "Z", 5.5, Ø
77Ø DATA "ORI", "Z", 5.65, -2
78Ø DATA "ORI", "Z", 5.8, -4
79Ø DATA "XXX", "XXX", Ø, Ø
8ØØ FOR J=12Ø TO 136 STEP 4
     : PSET(J, 9Ø, 1)
     : NEXT
81Ø FOR J=84 TO 98 STEP 4
     : PSET(128, J, 1)
     : NEXT
82Ø CIRCLE(128, 9Ø), 9Ø
83Ø DRAW "BM34, 6ØR4L4D2R2L2D2R4"
84Ø DRAW "BM226, 6Ø; D4E4F4U4"
85Ø READ F$, B$, A, B
86Ø IF B$="XXX" THEN 1ØØØ
87Ø A=24-A
88Ø T=DY/15+H
     : A=A+T+12
     : B=ABS(B-9Ø)
89Ø IF A>24 THEN A=A-24
     : GOTO 89Ø
9ØØ A=A*15
91Ø Y=SIN(A/57.296)*B+LA
92Ø X=128-(COS(A/57.296)*B)
93Ø IF Y<Ø OR Y>18Ø THEN GOTO 85Ø
94Ø IF X<38 OR X>218  THEN 99Ø
95Ø PSET(X, Y, 1)
96Ø IF B$<>"Z" THEN CIRCLE(X, Y), 1
97Ø IF F$=E$ THEN CIRCLE(X, Y), 1
98Ø IF B$=C$ THEN CIRCLE(A, B), 2
99Ø GOTO 85Ø
1ØØØ GOSUB 126Ø
     : GOSUB 111Ø
1Ø1Ø RESTORE
     : FOR J=1 TO 14
     : READ Z
     : NEXT
1Ø2Ø I$=INKEY$
     : IF I$= "Q" THEN  END
1Ø3Ø IF I$="E" THEN CIRCLE(128, LA), 9Ø, 1, 1,
     .Ø5, .45
     : GOTO 1Ø2Ø
1Ø4Ø IF I$="D" THEN RESTORE
     : GOTO 9Ø
```

```
105Ø IF I$="" THEN 1Ø2Ø
1Ø6Ø PCLS
     : H=H+1
     : GOTO 79Ø
1Ø7Ø IF H=7 OR H=17 THEN DRAW "BM2ØØ, 15Ø; R8G12"
1Ø8Ø IF H=8 OR H=18 THEN DRAW "BM2ØØ, 15Ø;
     R8D6L8D6R8U6 L6U6"
1Ø9Ø IF H=9 OR H=19 THEN DRAW "BM2ØØ, 15Ø;
     R8D12U6L8U6"
11ØØ IF H=1Ø OR H=2Ø THEN DRAW "BM2ØØ, 15Ø;
     R8D12L8U12"
111Ø IF H>24 THEN H=N-24
     : DY=DY+1
112Ø IF H=1 OR H=11 OR H=21 THEN DRAW "BM2ØØ,
     15Ø; D12"
113Ø IF H=2 OR H=12 OR H=22 THEN DRAW "BM2ØØ,
     15Ø; R8D2G6 D4R8"
114Ø IF H=3 OR H=13 OR H=23 THEN DRAW "BM2ØØ,
     15Ø; U2R8D6 L4R4D6L8U2"
115Ø IF H=4 OR H=14 OR H=24 THEN DRAW "BM2ØØ,
     15Ø; D6R12L6 U6D12"
116Ø IF H=5 OR H=15 THEN DRAW "BM2ØØ, 15Ø;
     R8L8D6R8D6L8 U2"
117Ø IF H=6 OR H=16 THEN DRAW "BM2ØØ, 15Ø;
     R6L6D12R8U6L6"
118Ø IF H=7 OR H=17 THEN DRAW "BM2ØØ, 15Ø; R8D12"
119Ø IF H=8 OR H=18 THEN DRAW "BM2ØØ, 15Ø;
     R8D6L8D6R8 U6L8U6"
12ØØ IF H=9 OR H=19 THEN DRAW "BM2ØØ, 15Ø;
     R6D6L6U6R8 D12"
121Ø IF H=1Ø OR H=2Ø THEN DRAW "BM2ØØ, 15Ø;
     R8D12L8U12"
122Ø IF H>9 AND H<2Ø THEN DRAW "BM18Ø, 15Ø; D12"
123Ø IF H>19 THEN DRAW "BM18Ø, 15Ø; R8D2G6D4R8"
124Ø SOUND 2ØØ, 8
125Ø RETURN
126Ø Y1=YY-82
127Ø DE=DY+9
     : IF DE>365 THEN DE=DE-365
128Ø DE=DE/57.296
     : E=COS(DE/365*36Ø)*23
129Ø A=H-6
     : IF H<6 OR H>18 THEN SCREEN 1, 1
     : GOTO 133Ø
13ØØ IF C$<>"SUN" THEN  133Ø
131Ø GOSUB 157Ø
132Ø SCREEN 1, Ø
     : CIRCLE(X, Y), 4
     : PAINT(X, Y), 1, 1
133Ø IF YY<82 OR YY>85 THEN RETURN
134Ø MM=M+Y1*12
135Ø FOR Z=1 TO MM
     : READ YA, MA, JA, SA
     : NEXT
136Ø IF YA=Ø THEN 14ØØ
137Ø A=VA/15+(H-6)
     : GOSUB 157Ø
     : GOSUB 163Ø
     : IF Y<LA  THEN  GOTO 139Ø
138Ø LINE(X, 182)-(X+2, 188), PSET
     :  LINE-(X+4, 182), PSET
139Ø IF MA=Ø THEN 143Ø
14ØØ A=MA+T
     : GOSUB 157Ø
     : GOSUB 163Ø
     : IF Y<LA THEN GOTO 142Ø
141Ø LINE(X, 188)-(X, 182), PSET
     : LINE-(X+2, 188),  PSET
     : LINE-(X+4, 182), PSET
     : LINE-(X+4, 188), PSET
142Ø IF JA=Ø THEN 146Ø
143Ø A=JA+T
     : GOSUB 157Ø
     : GOSUB 163Ø
     : IF Y<LA THEN GOTO 145Ø
144Ø LINE(X+2, 182)-(X+2, 188), PSET
     : LINE-(X-2, 188), PSET
     : LINE-(X-2, 184), PSET
145Ø IF SA=Ø THEN RETURN
146Ø A=SA+T
     : GOSUB 157Ø
     : GOSUB 163Ø
     : IF Y<LA THEN RETURN
147Ø LINE(X+2, 182)-(X-2, 182), PSET
     : LINE-(X+2, 188), PSET
     : LINE-(X-2, 188), PSET
148Ø RETURN
149Ø DATA Ø, 12, 8, 14, 4, 13, 4, 33, 13, 2,
     14.5, 13.4, 45,  13, 14.5, 13.3, 46, 12.4,
     14.3, 13.2, 42, 12.1, 14.1, 13.1, 36, 12.4,
     13.9, 13
15ØØ DATA 29, 13.2, 13.9, 13, 21, 13.9, 14.1,
     13.2, 13,  15.6, 14.4, 13.4, Ø, 17.1, 14.8,
     Ø, Ø, 18.8, Ø, 13.8, -11, 2Ø.4, 15.7, 14
151Ø DATA -18, 22, 16.1, 14.1, -25, 23.5, 16.5,
     14.2,  -31, .8, 16.6, 14.1, -31, 2.3, 16.6,
     14, -43, Ø, 16.4, 13.9, -45, Ø, 16.1, 13.8,
     -41, 6.8, 16, 13.8, -15, 8.2, 16, 13.9, 29,
     9.6, 16.2, 14
152Ø DATA 44, 1Ø.7, 16.5, 14.2, 46, 11.9, 16.9,
     14.5,  42, 13, Ø.14.7
153Ø DATA 37, 14, 17.9, 14.9, 31, 15, 18.4, 15,
     24, 15.6,  18.7, 15, 15, 15.7, 18.9, 14.9,
     Ø, 15.1, 18.9, 14.7, Ø, 14.6, 18.7, 14.6
154Ø DATA Ø, 14.8, 18.5, 14.5, -16, 15.7, 18.2,
     14.6,  -25, 17, 18.2, 14.7, -32, 18.5, 18.5,
     14.9, -39, 2Ø.1, 18.8, Ø, -44, 21.7, 19.3,
     15.4
155Ø DATA -47, 23.1, Ø, 15.6, -45, .6, 2Ø.3,
     15.7, -28,  1.8, 28.7, 15.8, 18, 3.3, 21,
     15.7, 41, 4.7, 21.3, 15.5, 46, Ø, 21.3, 15.4
156Ø DATA 43, Ø, 21.1, 15.3, 38, Ø, 2Ø.9, 15.3,
     3Ø, 1Ø.4,  2Ø.7, 15.4, 23, 11.1, 2Ø.7, 15.6,
     16, 12.7, 2Ø, 8, Ø, Ø, 13.9, 21.2, 16.1
157Ø B=9Ø
     : A=A*15
158Ø X=128-(COS(A/57.296)*B)
159Ø IF X<38 OR X>218 THEN RETURN
16ØØ Y=SIN(A/57.296)*B
161Ø Y=Y+LA+E
162Ø RETURN
163Ø IF X<38 OR X>218 THEN RETURN
164Ø IF Y<LA THEN RETURN
165Ø CIRCLE(X, Y), 1
     : IF C$="SUN" THEN  CIRLCE(X, Y), 2
     : RETURN
166Ø RETURN
```



Radio Shack Computer Center Addresses and Telephone Numbers

ALABAMA

BIRMINGHAM 2428 Green Springs Hwy., (205) 945-0792
HUNTSVILLE 1400 N. Memorial Pkwy., (205) 536-1581
MOBILE 405 Bel-Air Blvd., (205) 471-1617
MONTGOMERY #24 Union Square S/C, (205) 271-1500

ARIZONA

PHOENIX 10233 Metro Pkwy. E., (602) 861-1124; 4301 N. 7th Ave., (602) 277-3031
SCOTTSDALE 2525 N. Scottsdale Rd., (602) 990-2241
TEMPE 83 E. Broadway, (602) 894-2065
TUCSON 5622 E. Broadway, (602) 748-0101

ARKANSAS

LITTLE ROCK Town & Country S/C, University & Asher, (501) 568-5694

CALIFORNIA

ANAHEIM 509 Katella, (714) 776-9540
BAKERSFIELD 2018 Chester Ave., (805) 323-8074
BERKELEY 1922 Grove St., (415) 848-9170
BEVERLY HILLS 8500 Wilshire Blvd., (213) 659-8870
CANOGA PARK 8371 Topanga Canyon, (213) 347-9800
CARMICHAEL 6305 Fairoaks Blvd., (916) 484-6815
CHULA VISTA 1201 3rd Ave., (714) 420-3810
DOWNEY 8031 Florence Ave., (213) 927-7882
ESCONDIDO 347 W. Mission Ave., (714) 741-6032
FRESNO Princeton S/C, 2721 N. Blackstone Ave., (209) 225-5551
GARDEN GROVE 12821 Knott St., (714) 891-1160
GLENDALE 236 N. Brand Blvd., (213) 246-9310
HAYWARD 20942 Mission Blvd., (415) 278-2888
LAKEWOOD 5830 Lakewood Blvd., (213) 920-9671
LA MESA 5346 Jackson Dr., (714) 460-3610
LONG BEACH 2119 Bellflower Blvd., (213) 597-3377
LOS ANGELES 740 S. Olive St., (213) 629-2455
MONTCLAIR 5237 Arrow Hwy., (714) 625-1541
MONTEREY 484 Washington St., (408) 375-8430
MOUNTAIN VIEW 1933 El Camino Real W., (415) 961-0542
OAKLAND 1733 Broadway, (415) 763-3183
PASADENA 575 S. Lake Ave., (213) 449-5424
RIVERSIDE 3844 La Sierra Ave., (714) 689-0340
SACRAMENTO 4749 J. St., (916) 454-3287
SAN BERNARDINO 764 Inland Center Dr., (714) 884-6871
SAN DIEGO 3062 Clairemont Dr., (714) 276-6050; 3902 El Cajon Blvd., (714) 280-0227
SAN FRANCISCO One Market Place, (415) 777-9810
SAN JOSE 1228 S. Bascom Ave., (408) 297-2603
SAN MATEO 3180 Campus Dr., (415) 573-8607
SANTA BARBARA 4141 State St. A-1, (805) 967-4538
SANTA FE SPRINGS 14138 East Firestone Blvd., (213) 921-0702
STOCKTON College Sq. S/C, 963 West March Lane, (209) 957-3676
TARZANA 18545 Ventura Blvd., (213) 343-1696
TORRANCE 3840 Sepulveda at Hawthorne, (213) 373-0306
VENTURA 4005 E. Main St., (805) 654-0196
WEST COVINA 2516 E. Workman St., (213) 915-5791

COLORADO

BOULDER Arapahoe Plaza, 3550 Arapahoe, (303) 443-7142
COLORADO SPRINGS 4341 N. Academy, (303) 593-7500
DENVER 8000 E. Quincy, (303) 770-1362
LAKEWOOD 2099 Wadsworth Blvd., (303) 232-6277

CONNECTICUT

EAST HAVEN 51 Frontage Rd., (203) 467-8864
FAIRFIELD 1196 Kings Hwy. & Rt. 1, (203) 255-6099
HARTFORD The Richardson Bldg., 942 Main St., (203) 549-5717
MANCHESTER 228 Spencer St., (203) 649-8210
NORWALK Rt. 7-345 Main Ave., (203) 846-3418
ORANGE Woolco S/C; 538 Boston Post Rd., (203) 795-1291
WATERBURY 105 Bank St., (203) 573-8800
WATERFORD 122 Boston Port Rd., (203) 443-0716
WEST HARTFORD 39 S. Main St., (203) 523-4283

DELAWARE

WILMINGTON 3847 Kirkwood Hwy., (302) 999-0193

DISTRICT OF COLUMBIA

WASHINGTON 1800 M St. NW., (202) 822-3933

FLORIDA

ALTAMONTE SPRINGS 766 B. East Altamonte Dr., (305) 339-1313
CLEARWATER 2460-D US 19 North, (813) 797-3223
DAYTONA BEACH Volusia Plaza, (904) 255-5909
FT. LAUDERDALE 4368 N. Federal Hwy., (305) 772-6797
GAINESVILLE 3315 Archer Rd., (904) 375-7063
HOLLYWOOD 429 S. State Rd. #7, (305) 966-4382
JACKSONVILLE 8252 Arlington Exprwy., (904) 725-2594
LAUDERDALE LAKES 4317-25 N. State Rd. 7, (305) 486-2240
MIAMI 9459 S. Dixie Hwy., (305) 667-2316; 1601 Biscayne Blvd., (305) 374-6433; 15 SE. 2nd Ave., (305) 374-7310; 20761 S. Dixie Hwy., (305) 238-2518
N. MIAMI BEACH The Promenade, 1777 N.E. 163rd St., (305) 940-6887
ORLANDO 1238 E. Colonial Dr., (305) 894-0570
ST. PETERSBURG 3451 66th St. N., (813) 381-2366
SARASOTA 5251 S. Tamiami Tr. (Hwy.41), (813) 923-4721
TALLAHASSEE 2529 S. Adams, (904) 222-4440
TAMPA 4555 W. Kennedy, (813) 879-7470; 1825 E. Fowler Ave., (813) 971-1130
W. PALM BEACH 2271-A Palm Beach Lakes Blvd., (305) 683-3100

GEORGIA

AUGUSTA 3435 Wrightsboro Rd., (404) 738-5998
ATLANTA 2108 Henderson Mill, (404) 939-9888; 49 W. Paces Ferry, (404) 231-9604; Akers Mill S/C, 2937 Cobb Parkway NW., (404) 955-5235; 113 Peachtree St., (404) 223-5904
COLLEGE PARK 5309 Old National Hwy., (404) 761-3055
DORAVILLE 5697 Buford Hwy., (404) 458-2691
SAVANNAH Chatham Plaza, 7805 Abercorn St., (912) 355-6074

IDAHO

BOISE 691 S. Capitol Blvd., (208) 344-5450

ILLINOIS

AURORA 890 North Lake St., (312) 844-2224
CHICAGO 4355 S. Archer Ave., (312) 376-7617; CNA Plaza, 309 S. Wabash, (312) 922-0536
ELMWOOD PARK 7212 W. Grand Ave., (312) 452-7500
FAIRVIEW HEIGHTS #4 Market Place, (618) 398-6410
HOMEWOOD/GLENWOOD 329 Glenwood Lansing, (312) 758-0440
LaGRANGE One S. LaGrange Rd., (312) 482-3484
LIBERTYVILLE 1350 S. Milwaukee Ave., (312) 367-8230
LOMBARD 4 Yorktown Center, (312) 629-5350
NILES 8349 Golf Rd., (312) 470-0670
OAK LAWN 4815 W. 95th St., (312) 425-9130
PEORIA 4125 N. Sheridan Rd., (309) 685-7056
ROCKFORD North Town S/C, 3600 N. Main St., (815) 282-1001
SCHAUMBURG 651 Mall Dr., (312) 884-8600
SPRINGFIELD Sherwood Plaza, 2478 Wabash, (217) 787-3066

INDIANA

EVANSVILLE 431 Diamond Ave., (812) 426-1715
FT. WAYNE 747 Northcrest S/C, (219) 482-9547
GRIFFITH 208 W. Ridge Rd., (219) 838-3000
INDIANAPOLIS 6242 E. 82nd St., Castleton Plz., (317) 849-6896; Speedway Plaza, 6129 B Crawfordsville, (317) 244-2221; 10013 E. Washington St., (317) 898-4887
TERRE HAUTE 3460 U.S. Hwy. 41 S., (812) 234-3212

IOWA

DAVENPORT 616 E. Kimberly Rd., (319) 386-3457
DES MOINES 7660 Hickman Rd., Sherwood Forest S/M, (515) 270-0193

KANSAS

OVERLAND PARK 8619 W. 95th, (913) 642-1301
TOPEKA White Lakes Plaza, West Tower, 3715 Plaza Dr., (913) 267-6420
WICHITA 2732 Blvd. Plaza S/C, (316) 681-1212

KENTUCKY

FLORENCE 7727 Mall Rd., (606) 371-2811
LEXINGTON 2909 Richmond Rd., (606) 269-7321
LOUISVILLE 2900 Taylorsville Rd., (502) 459-9901

LOUISIANA

BATON ROUGE 7007 Florida Blvd., (504) 928-5260
METAIRIE 3750 Veterans Hwy., (504) 454-3681
NEW ORLEANS 327 St. Charles Ave., (504) 523-6408
SHREVEPORT 1545 Line Ave., (318) 221-5125

MAINE

BANGOR Maine Square, (207) 945-6491

MARYLAND

BALTIMORE 7942 Belair Rd., Putty Hill Plaza, (301) 882-9583; 115 N. Charles St. at Lexington, (301) 539-7251
CATONSVILLE One Mile West S/C, 6600 B Balt. Nat'l. Pike, (301) 788-3277
FREDERICK Shoppers World, Rt. 40W, (301) 695-8440
NEW CARROLLTON-LANHAM 7949 Annapolis Rd., (301) 459-8030
PASADENA 8120 Ritchie Hwy., (301) 544-2352
ROCKVILLE Congressional Plaza, 1673 Rockville Pike, (301) 984-0424
SALISBURY Shoppers World S/C, Rt. 50, (301) 546-9223

MASSACHUSETTS

BOSTON 730 Commonwealth Ave., (617) 739-1704; 111 Summer St., (617) 542-0361
BRAINTREE South Shore Plaza, 250 Granite St., (617) 848-9290
BROCKTON 675 Belmont, (617) 583-2270
BURLINGTON Crossroads Plaza, Rt. 3 S., (617) 229-2850
CAMBRIDGE Harvard Square, 28 Boylston St., (617) 354-7694
CHESTNUT HILL 200 Boylston St., (617) 969-2031
NATICK 1400 Worcester Rd., (617) 875-8721
SAUGUS 343 Broadway, (617) 233-4985
SPRINGFIELD 1985 Main St., Northgate Plz., (413) 732-4745
WORCESTER Lincoln Plaza, (617) 852-8844

MICHIGAN

BIRMINGHAM 3620 W. Maple Rd., (313) 647-2151
DETROIT DWNTN 1559 Woodward Ave., (313) 961-6855
FLINT G3298 Miller Rd., Yorkshire Plaza, (313) 732-2530
GRAND RAPIDS 3142 28th St. SE., (616) 957-2040
KALAMAZOO 25 Kalamazoo Center, (616) 343-0780
LANSING 2519 S. Cedar St., (517) 372-1120
LIVONIA 33470 W. 7 Mile Rd., (313) 476-6800
ROSEVILLE 31873 Gratiot Ave., (313) 296-6210
SOUTHFIELD 17651 West 12 Mile Rd., (313) 569-1027
TROY Oakland Plaza, 322 John R. Rd., (313) 585-3900

MINNESOTA

BLOOMINGTON 10566 France Ave. S., (612) 884-1641
GOLDEN VALLEY Golden Valley S/C, 8016 Olson Memorial Hwy., (612) 542-8471
ST. PAUL 6th & Wabasha, (612) 291-7230

MISSISSIPPI

JACKSON 979 Ellis Ave., (601) 352-5001

MISSOURI

FLORISSANT 47 Florissant Oaks S/C, (314) 921-7722
INDEPENDENCE 1325 S. Noland Rd., (816) 254-3701
KANSAS CITY 4025 N. Oak Trafficway, (816) 455-3381
ST. ANN 10472 St. Charles Rock Rd., (314) 428-1400
SPRINGFIELD 2684 S. Glenstone, (417) 883-4320

NEBRASKA

OMAHA 3006 Dodge St., (402) 346-4003

NEVADA

LAS VEGAS Commercial Center, 953 E. Sahara #31-B, (702) 731-3956
RENO 3328 Kietzke Lane, (702) 826-6327

NEW HAMPSHIRE

MANCHESTER Hampshire Plaza, 1000 Elm St., (603) 625-4040

NEW JERSEY

BRIDGEWATER 1472 U.S. Highway 22 East, (201) 469-3232
E. BRUNSWICK 595 A Rt. 18, (201) 238-7142
E. HANOVER Rt. 10, Hanover Plaza, (201) 884-1200
LAWRENCEVILLE Rt. 1 & Texas Ave., (609) 771-8113
NEWARK 595 Broad, (201) 622-1339
PARAMUS 175 Rt. 17 S., (201) 262-1920
SPRINGFIELD Rt. #22 Center Isle, (201) 467-9827
VOORHEES 35 Eagle Plaza, (609) 346-0600

NEW MEXICO

ALBUQUERQUE 2108 San Mateo NE., (505) 265-9587

NEW YORK

ALBANY Shoppers Pk., Wolf Rd., (518) 459-5527
BAYSHORE 1751 Sunrise Hwy., (516) 666-1800
BETHPAGE 422 N. Wantagh Ave., (516) 822-6403
BUFFALO 839 Niagara Falls Blvd., (716) 837-2590
FRESH MEADOWS 187-12 Horace Harding Exp., (212) 454-1075
JOHNSON CITY Giant Shopping Center, Harry L. Drive, (607) 729-6312
MELVILLE TSS Mall, Rt. 110, (516) 673-4646
NEWBURGH Zayre Plaza, Rt. #17K, (914) 561-2960
NEW ROCHELLE 242 North Ave., (914) 636-0700
NEW YORK 385 Fifth Ave., (212) 889-1345; 139 E. 42nd St., (212) 953-6053; 19 W. 23rd St., (212) 691-1861; 347 Madison Ave., (212) 867-8650
REGO PARK 97-77 Queens Blvd., (212) 897-5200
ROCHESTER 3000 Winton Rd., (716) 244-6400
SCARSDALE 365 Central Park Ave., (914) 472-2520
SPRING VALLEY White House Center, 88 W. Rt. 59, (914) 425-2828
STATEN ISLAND 2409 Richmond Ave., (212) 698-3100
SYRACUSE 2544 Erie Blvd., (315) 446-3017; Hotel Syracuse, 510 S. Warren St., (315) 471-6663
UTICA Riverside Mall, (315) 735-1933

NORTH CAROLINA

CHARLOTTE 3732 Independence Blvd., (704) 535-6320
GREENSBORO 3718 High Point Rd., (919) 294-5529
RALEIGH Townridge Sq., Hwy. 70 W., (919) 781-9380
WINSTON-SALEM 629 Peters Creek Pkwy., (919) 722-0030

OHIO

AKRON Fairlawn Plaza, 2727 W. Market St., (216) 836-9303
CANTON 5248 Dressler Rd. NW., (216) 494-7230; Mellet Plaza, 3826 W. Tuscarawas, (216) 478-1878
CENTERVILLE 2026 Miamisburg-Centerville Rd., (513) 435-5167
CINCINNATI 9725 Montgomery, (513) 793-8688
CLEVELAND 419 Euclid (Dwntwn), (216) 575-0800; 27561 Euclid Ave., (216) 289-6823
COLUMBUS 862 S. Hamilton, Great Eastern S/C, (614) 864-2806; The Patio Shop. Ctr., 4661 Karl Rd., (614) 436-4666; 400 N. High St., (614) 464-2781
DAYTON Northwest Plaza, 3279 West Siebenthaler, (513) 277-6500
NORTH OLMSTED Great Northern S/C, (216) 734-2255
PARMA 7551 W. Ridgewood Dr., (216) 842-4030
TOLEDO 5844 W. Central Ave., (419) 531-5797
YOUNGSTOWN Union Square Plaza, 2543 Belmont Ave., (216) 744-4541

OKLAHOMA

OKLAHOMA CITY 4732 SE 29th St., (405) 670-4561; Springdale S/C, 4469 NW 50th, (405) 943-8712; 1101 SW 59th St., (405) 634-2406
TULSA 7218 & 7220 E. 41st St., (918) 663-2190

OREGON

EUGENE 390 Coburg Rd., (503) 687-0082
PORTLAND 7463 SW Barbur Blvd., (503) 246-1157; 9131 SE Powell, (503) 777-2223

PENNSYLVANIA

ALLENTOWN Crest Plaza S/C, Cedar Crest Blvd. US 22, (215) 395-7155
BALA CYNWYD 67 E. City Line Ave., (215) 668-9950
ERIE 5755 Peach St., (814) 868-5541
HARRISBURG Union Deposit Mall, Union Deposit Rd. #17, (717) 564-6753
LANCASTER Park City Plaza, US 30, (717) 393-5817
MONROEVILLE 3828 Wm. Penn. Hwy., (412) 823-3400
MONTGOMERYVILLE Airport Sq., Rt. 309, (215) 362-1200
PHILADELPHIA 7542 Castor Ave., (215) 342-2217; 1002 Chestnut St., (215) 923-3080
PITTSBURGH 5775 Baptist Rd., Hills Plaza, (412) 831-9694; 303 Smithfield St., (412) 391-3150
SCRANTON 206 Meadow Ave., (717) 348-1801

RHODE ISLAND

E. PROVIDENCE 850 Waterman Ave., (401) 438-2860
PROVIDENCE 177 Union St., (401) 272-0201

SOUTH CAROLINA

COLUMBIA Old Sears Bldg., 1001 Harden St., (803) 799-2065
GREENVILLE N. Hills S/C, (803) 292-1835
N. CHARLESTON 5900 Rivers Ave., (803) 747-5580

TENNESSEE

CHATTANOOGA 636 Northgate Mall, (615) 870-1366
JOHNSON CITY Peerless Center, (615) 282-6829
KNOXVILLE Cedar Bluff S/C, 9123 Executive Park Dr., (615) 690-0520
MEMPHIS 4665 American Way, (901) 795-4963; 1997 Union Ave., (901) 278-7935
NASHVILLE 2115 Franklin Pike, (615) 298-5484; Rivergate Plaza, (615) 859-3414

TEXAS

AMARILLO Wellington Sq. S/C, 1619 S. Kentucky, (806) 358-4567
ARLINGTON 2500 E. Randol Mill, Suite 113, (817) 274-3127
AUSTIN 8764 E. Research Blvd., (512) 459-4238
BROWNSVILLE 1639 Price Rd. (Hwy. 77), (512) 544-6800
BEAUMONT 5330 Eastex Frwy., (713) 898-7000
CORPUS CHRISTI 1711 S. Staple St., (512) 887-8901
DALLAS 15340 Dallas Pkwy., Suite 1100, (214) 934-0275; 2930 W. Northwest Hwy., (214) 350-4144; 1517 Main St., (214) 760-8601; 2588 Royal Ln., (214) 484-9947
EL PASO 9515 Gateway West, (915) 594-8211
FT. WORTH 15 One Tandy Center, (817) 335-7198; 2801 Alta Mere, (817) 738-0251
HARLINGEN 1514 S. Hwy 77, Sunshine Strip, (512) 425-8890
HOUSTON 211C-FM 1960, (713) 444-7006; 10543 Gulf Fwy., (713) 943-9310; 5900 North Fwy., (713) 699-1932; 6813 SW Fwy., (713) 777-7907; 809 Dallas St., (713) 651-3002
HURST Northeast Mall, (817) 284-1518
LUBBOCK 3625 34th St., (806) 793-1467
ODESSA 1613 "A" East 8th Street, (915) 334-8355
RICHARDSON Fleetwood Sq. S/C, 202 W. Campbell Rd., (214) 669-1494
SAN ANTONIO 6018 West Ave., (512) 344-8792; 4249 Centergate, (512) 657-3958

UTAH

MURRAY 6051 S. State Ave., (801) 268-8978
SALT LAKE CITY 415 5th Ave., (801) 322-4893

VIRGINIA

ALEXANDRIA 3425 King St. at Quaker Ln., (703) 370-9000
FAIRFAX Westfair Center, 11027 Lee Hwy., (703) 273-6500
NEWPORT NEWS Newmarket South Shop. Ctr., (804) 245-8203
NORFOLK 5731 Poplar Hall Dr., (804) 461-0798
RICHMOND Willow Lawn S/C, 1617 Willow Lawn Dr., (804) 282-3453; 7728 Midlothian Turnpike, (804) 272-8803
ROANOKE Franklin Bldg., 3561 Franklin Rd. S.W., (703) 342-6335
ROSSLYN 1911 N. Ft. Myer Dr. at Rt. 29, (703) 243-5955

WASHINGTON

BELLEVUE Crossroads Mall, North East 8th & 156 St., (206) 644-1804
BELLINGHAM 1111 Cornwall Ave., Suite B & C, (206) 676-0811
FEDERAL WAY 33505 Pacific Hwy. South, (206) 838-6830
SEATTLE 18405 Aurora Ave. N., (206) 542-6184; 1521 3rd Ave., (206) 447-1959; 5030 Roosevelt Way NE, (206) 527-1019
SPOKANE 7702 N. Division, (509) 484-7000; E. 12412 Sprague, (509) 922-2800
TACOMA 7030 S. Sprague, (206) 473-7333
TUKWILA 15425 53rd Ave. S., (206) 248-3710
YAKIMA 1111 N. First St., (509) 248-9667

WEST VIRGINIA

HUNTINGTON 2701½ 5th Ave., (304) 523-3527

WISCONSIN

MADISON 57 West Towne Mall, (608) 833-6130
MILWAUKEE 6450 N. 76th St., (414) 353-6790
WEST ALLIS 2717 South 108th St., (414) 327-4240

Radio Shack

TRS-80 Microcomputer News
P.O. Box 2910
Fort Worth, Texas 76113-2910

FIRST CLASS
U.S. POSTAGE
PAID
DALLAS, TX
PERMIT 2180

ADDRESS CHANGE
☐ **Remove from list** ☐ **Change as shown**
Please detach address label and mail to address shown above